The Power of Randomized Algorithms: From Numerical Linear Algebra to Biological Systems

by

Cameron Nicholas Musco

B.S., Computer Science, Yale University (2012)
B.S., Applied Mathematics, Yale University (2012)
S.M., Computer Science, Massachusetts Institute of Technology (2015)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2018

Author . . . .
Department of Electrical Engineering and Computer Science
June 22, 2018

Certified by . . . .
Nancy A. Lynch
NEC Professor of Software Science and Engineering
Professor of Electrical Engineering and Computer Science
Thesis Supervisor

Accepted by . . . .
Leslie Kolodziejski
Professor of Electrical Engineering and Computer Science
Chair, Department Committee on Graduate Theses

The Power of Randomized Algorithms: From Numerical Linear Algebra to Biological Systems

by

Cameron Nicholas Musco

Submitted to the Department of Electrical Engineering and Computer Science
on June 22, 2018, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Electrical Engineering and Computer Science

Abstract

In this thesis we study simple, randomized algorithms from a dual perspective. The first part of the work considers how randomized methods can be used to accelerate the solution of core problems in numerical linear algebra. In particular, we give a randomized low-rank approximation algorithm for positive semidefinite matrices that runs in sublinear time, significantly improving upon what is possible with traditional deterministic methods. We also discuss lower bounds on low-rank approximation and spectral summarization problems that attempt to explain the importance of randomization and approximation in accelerating linear algebraic computation.

The second part of the work considers how the theory of randomized algorithms can be used more generally as a tool to understand how complexity emerges from low-level stochastic behavior in biological systems. We study population density-estimation in ant colonies, which is a key primitive in social decision-making and task allocation. We define a basic computational model and show how agents in this model can estimate their density using a simple random-walk-based algorithm. We also consider simple randomized algorithms for computational primitives in spiking neural networks, focusing on fast winner-take-all networks.

Thesis Supervisor: Nancy A. Lynch
Title: NEC Professor of Software Science and Engineering
Professor of Electrical Engineering and Computer Science

Acknowledgments

First, I would like to thank my advisor Nancy Lynch. As the contents of this thesis evidence, my research has spanned a range of areas, which Nancy has been happy to let me wander into. At the same time, she has been a hands-on advisor when needed, driving much of my work on biological computation and teaching me how to build a new research area, which is as much about proposing new kinds of problems as about solving those problems.

I would also like to thank David Woodruff and Piotr Indyk, the other members of my thesis committee. I was fortunate to have interned with David at IBM Research. Since then he has been both a frequent collaborator and a valuable mentor to me. Piotr has always been a friendly face in the theory group and a willing source for advice on research and beyond.

I would also like to thank my many collaborators, from whom I have received the vast the majority of my education as a Ph.D. student. I am especially grateful to Aaron Sidford, whom I began working with in my first year at MIT and has been a invaluable guide and friend. I am also grateful to Michael Kapralov, Mira Radeva, Merav Parter, and Babis Tsourakakis, who have taught me so much throughout our work together.

While we have never written a paper together, I would also like to thank Jelani Nelson, Michael Mahoney, and Petros Drineas, who have closely followed my research and provided mentorship and support throughout my Ph.D. I would also like to thank Joanne Hanley for all her support and encouragement throughout my time at MIT.

Thanks so much to my family: Mom, Dad, Christopher, Brett, Jenna, Nicole, Tyler, Amanda, and all the extended family; and to Christina. They are everything to me and are the support behind everything that I do. Christopher and I have operated as a team literally since day one, and he has been the greatest support as we have embarked on our Ph.D.'s together.

Finally, I would like to thank Michael Cohen, who was a close collaborator and friend, that sadly passed away of an illness last year. It is impossible for me to describe Michael fully in words. He was by far the most brilliant person I have ever encountered, and approached the world with unmatched openness, excitement, humbleness, and happiness that we can only seek to emulate. His impact on me is one of the greatest things I will take away from my time at MIT.

Contents

1 Introduction **13**

1.1 A Dual View of Randomized Computation 14

1.2 This Thesis 15

1.2.1 Sublinear Time Low-Rank Approximation 15

1.2.2 Lower Bounds for Linear Algebraic Computation 17

1.2.3 Ant Colony Density Estimation 19

1.2.4 Computation in Spiking Neural Networks 20

1.3 Notation and Preliminaries 21

1.3.1 General Notation 22

1.3.2 Matrix and Vector Notation 22

2 Sublinear Time Low-Rank Approximation of PSD Matrices **25**

2.1 Background and Introduction to Results 26

2.1.1 Low-rank Approximation of PSD Matrices 28

2.1.2 Our Contributions 29

2.1.3 Algorithm Overview 30

2.1.4 Some Further Intuition on Error Guarantees 36

2.1.5 Road Map 37

2.2 Ridge Leverage Score Sampling 38

2.2.1 Leverage Score Definitions and Basic Properties 38

2.2.2 Approximation Bounds 41

2.2.3 Fast Ridge Leverage Score Approximation 51

2.3 Column Sampling 52

2.4 Row Sampling 55

2.4.1 Approximating the Ridge Leverage Scores of $\mathbf{AS}_1$ 55

2.4.2 Projection-Cost-Preserving Row Sampling 56

2.4.3 Spectral Norm Projection-Cost-Preservation 58

2.4.4 Frobenius Norm Projection-Cost-Preservation 65

2.5 Full Low-Rank Approximation Algorithm 71
2.5.1 Basic Algorithm 71
2.5.2 Outputting a PSD Matrix 82
2.6 Spectral Norm Error Bounds 85
2.6.1 Algorithmic Approach 86
2.6.2 Basic Algorithm 87
2.6.3 Sublinear Time Ridge Regression 91
2.7 Query Lower Bound 94
2.7.1 Lower Bound Approach 94
2.7.2 Primitive Approximation 95
2.7.3 Lower Bound for Low-Rank Approximation 97
2.8 Low-Rank Approximation of $\mathbf{A}$ via $\mathbf{A}^{1/2}$ 101
2.8.1 Converting a Low-Rank Approximation of $\mathbf{A}^{1/2}$ to a Low-Rank Approximation of $\mathbf{A}$ 101
2.8.2 PSD Low-Rank Approximation in $n^{1.69} \cdot \text{poly}(k/\epsilon)$ Time . . . 103
2.9 Discussion and Future Work 105
2.9.1 Sublinear Time Algorithms for PSD Matrices 105
2.9.2 Sublinear Time Algorithms for Other Matrix Types 106
2.9.3 Expanding the Applications of Leverage Scores 107

3 Lower Bounds for Linear Algebraic Computation **109**
3.1 Background and Introduction to Results 110
3.1.1 Spectral Sum Problems 110
3.1.2 Kernel Low-Rank Approximation 113
3.1.3 Our Contributions 116
3.1.4 Prior Work 120
3.2 Lower Bounds for Spectrum Approximation 120
3.2.1 Lower Bound Approach 121
3.2.2 Reductions From Triangle Detection 123
3.2.3 Hardness for Computing Spectral Sums 126
3.2.4 Leverage Score and Effective Resistance Hardness 130
3.2.5 Determinant Hardness 134
3.3 Lower Bounds for Kernel Approximation 136
3.3.1 Lower Bound Approach 136
3.3.2 Lower Bound for Gram Matrices 137
3.3.3 Lower Bound for Dot Product Kernels 140

3.3.4 Lower Bound for Distance Kernels 143
3.3.5 Fast Low-Rank Approximation of $\mathbf{AA}^T$ 146
3.3.6 Hardness of Outputting a Low-Rank Subspace 147
3.4 Discussion and Future Work 148
3.4.1 Connecting Matrix Multiplication to Other Problems 148
3.4.2 Understanding the Role of Randomness 149

4 Ant-Inspired Density Estimation 151
4.1 Background and Introduction to Results 152
4.1.1 Density Estimation on the Grid 152
4.1.2 Our Contributions 153
4.1.3 Road Map 154
4.2 Theoretical Model for Density Estimation 155
4.2.1 Computational Model 155
4.2.2 The Density Estimation Problem 156
4.3 Random-Walk-Based Density Estimation 157
4.3.1 Random-Walk-Based Density Estimation Analysis 158
4.3.2 Decomposition of Collision Count into Independent Random Variables 158
4.3.3 Correctness of Encounter Rate in Expectation 159
4.3.4 A Re-collision Probability Bound 160
4.3.5 Collision Moment Bound 167
4.3.6 Correctness of Encounter Rate With High Probability 172
4.4 Extensions to Other Regular Topologies 175
4.4.1 From Re-collision Bounds to Density Estimation 175
4.4.2 Density Estimation on the Ring 176
4.4.3 Density Estimation on k-Dimensional Tori 179
4.4.4 Density Estimation on Regular Expanders 180
4.4.5 Density Estimation k-Dimensional Hypercubes 181
4.5 Independent-Sampling-Based Density Estimation 184
4.6 Applications 186
4.6.1 Social Network Size Estimation 186
4.6.2 Distributed Density Estimation by Robot Swarms 196
4.7 Discussion and Future Work 196
4.7.1 Extensions to Our Model 197
4.7.2 Biological Applications 198

4.7.3 Algorithmic Applications 199

5 Computation in Spiking Neural Networks 201

5.1 Background and Introduction to Results 202

5.1.1 Spiking Neural Networks 202

5.1.2 The Winner-Take-All Problem 203

5.1.3 Our Contributions 204

5.1.4 Road Map 205

5.2 Spiking Neural Network Model 206

5.2.1 Network Structure 206

5.2.2 Network Dynamics 208

5.2.3 Problems and Solving Problems 209

5.2.4 Basic Results and Properties of the Model 210

5.2.5 Potential Modifications to the Basic Model 215

5.2.6 The Winner-Take-All Problem 216

5.3 A Two-Inhibitor Solution to the WTA Problem 218

5.3.1 Network Definition 218

5.3.2 Basic Results and One-step Lemmas 221

5.3.3 Stability 227

5.3.4 Convergence to Good Configurations 228

5.3.5 Transition Lemmas for Good Configurations 232

5.3.6 Convergence to WTA 238

5.3.7 Completing the Bounds 246

5.4 WTA Lower Bounds 249

5.4.1 Single Auxiliary Neuron Lower Bound 251

5.4.2 Two Auxiliary Neuron Lower Bound 259

5.5 Faster Convergence With More Inhibitors 276

5.5.1 Use of History Period 276

5.5.2 $O(1)$ Convergence Time with $O(\log n)$ Inhibitors 279

5.5.3 Two-Step Lemmas 282

5.5.4 Stability 291

5.5.5 Convergence in $O(1)$ Steps 294

5.5.6 Completing the Analysis 309

5.5.7 Constructions With Runtime Tradeoffs 314

5.6 Discussion and Future Work 316

5.6.1 Winner-Take-All Extensions and Open Questions 317

5.6.2 Other Neural Computational Primitives 319
5.6.3 Learning Problems and Dynamic Networks 321
5.6.4 Neural Linear Algebraic Computation 322

Chapter 1

Introduction

It is well understood that randomness is a powerful computational tool. Randomization seems to aid in the solution of many problems, from important questions related to complexity theory like polynomial identity testing [Sax09], to basic communication complexity problems like equality testing [ES02], to fundamental graph problems like cut and flow approximation [BK96, She13]. In numerical linear algebra and data analysis, randomization is used to initialize iterative eigenvector algorithms [KW92, MM15], to seed clustering algorithms [AV07], to rapidly summarize inputs to accelerate computation [Sar06, CW13, NN13, MM13], and in stochastic optimization algorithms, like stochastic gradient descent, which are the methods of choice for training complex models like deep neural networks [LBH15].

In these applications and many others, randomized algorithms can give much faster, lower memory, and lower communication solutions than deterministic methods, sometimes with a provable gap. Aside from improvements in these traditional complexity measures, randomized algorithms also often seem to come with other advantages in both theory and in practical applications. They can aggressively trade approximation for efficiency and are often robust to noise and perturbations. They often take simple and inexpensive 'local' steps that access a small portion of the input. For this reason, they can often be naturally adapted to low-memory streaming or distributed environments. Finally, randomized methods can often be rigorously analyzed as probabilistic processes, giving provable runtime and approximation guarantees for worst-case inputs.

1.1 A Dual View of Randomized Computation

Throughout my Ph.D. I have explored randomized computation from two different perspectives:

1. An *engineering perspective* – how can we use randomization and approximation to design faster algorithms? In particular I have focused on basic problems in computational linear algebra and data analysis like linear regression [CLM+15, FMMS16], low-rank matrix approximation [MM15, GHJ+16, CMM17, MW17b], clustering [CEM+15], kernel learning [MM17, AKM+17], and spectral summarization [MNS+18]. We have employed randomization to give significant speedups over traditional algorithms, such as iterative and direct methods in numerical linear algebra, both in terms of theoretical runtime bounds and practical performance.

2. A *scientific perspective* – how can the analysis of randomized algorithms be used to study systems in which complexity emerges from low-level stochastic behavior? In particular I have focused on biological systems, including social insect colonies and spiking neural networks. We model these systems as computational processes and attempt to understand how their behaviors are driven by randomized interactions between many simple, distributed 'processors'. Our research builds on work studying a wide variety of biological phenomena from a computational viewpoint, including bird flocks [Cha09], cellular differentiation [AAB+11], evolution [LPDF08], and optimization by slime molds [BMV12].

In this thesis, we highlight four sets of results, two that study fast randomized algorithms for linear algebraic problems and two that study randomized computation in biological systems.

- In Chapter 2 we study random sampling algorithms for fast low-rank approximation of positive semidefinite (PSD) matrices. We show that random sampling can give sublinear time algorithms for general PSD matrices, going significantly beyond what is possible with both traditional deterministic methods and random-projection-based methods.

- In Chapter 3 we prove lower bounds that attempt to explain the importance of randomization and approximation in accelerating linear algebraic computation. In particular, we show that approximation seems to be necessary to give fast algorithms for a number of important problems including determinant, trace inverse, nuclear norm, and trace exponential computation. We also show that existing randomized algorithms for low-rank kernel matrix approximation are near-optimal, barring a breakthrough in fast matrix multiplication.

- In Chapter 4 we study how ant colonies estimate population density, a quantity used for a variety of social decision making, task allocation, and optimization tasks. We show how population density can be estimated using a simple algorithm based on tracking the number of times that ants encounter each other while random walking on a two-dimensional surface. We discuss applications of related algorithms beyond biology, including to random-crawl-based network size estimation.

- In Chapter 5 we study computation in stochastic spiking neural networks, and, in particular, consider the complexity of the classic winner-take-all symmetry breaking problem. We show that this problem can be solved efficiently using a very simple two-auxiliary neuron network that employs a random-competition-based strategy. We also give lower bounds demonstrating the near-optimality of this network, along with constructions that trade off a larger network size for a faster convergence time.

We conclude each chapter with a review of future directions and open questions. The landscape of randomized methods for numerical linear algebra, optimization, learning, and estimation is still wide open, and there is significant potential to use randomness to give faster algorithms for fundamental problems. In the area of biological computation, our work represents a few preliminary steps. There are many modifications to the simple models we consider that are worth studying and many computational problems solved by biological systems that we have not yet considered.

1.2 This Thesis

We now give an overview of the contents of each chapter of this thesis.

1.2.1 Sublinear Time Low-Rank Approximation

In Chapter 2 we highlight an application of random sampling methods to the problem of computing a low-rank approximation of a positive semidefinite (PSD) matrix, initially published in [MW17b]. Positive semidefinite matrices comprise an important class of matrices, including, for example, covariance matrices, graph Laplacians, and Gram matrices (in particular, kernel matrices).

We demonstrate that, by approximating a PSD matrix with a small random subset of columns and rows sampled according to their *ridge leverage scores*, a near-optimal low-rank approximation can be computed in sublinear time. Specifically, computing a rank-k approximation to an $n \times n$ PSD matrix achieving error within a $(1+\epsilon)$ factor of the optimal can be done in $\tilde{O}(n \cdot \text{poly}(k/\epsilon))$ time, even though the matrix may have up to n^2 nonzero entries. Our result represents a significant improvement from what is possible with traditional deterministic methods for low-rank approximation, which require $\Omega(n^3)$ time[1] and other randomized methods, like random projection, which require $\Omega(\text{nnz}(\mathbf{A}))$ time [CW13], where $\text{nnz}(\mathbf{A})$ denotes the number of nonzero entries in $\mathbf{A}$, and may be $\Omega(n^2)$ for dense inputs.

Our sublinear time algorithm starts with the observation that any PSD matrix $\mathbf{A}$ can be written as $\mathbf{B}^T\mathbf{B}$ for some matrix $\mathbf{B}$. That is, the entries of $\mathbf{A}$ correspond to the pairwise dot products between the columns of $\mathbf{B}$. This simple fact places a number of geometric constraints on these entries, which can be leveraged to achieve sublinear time low-rank approximation.

We can implicitly take advantage of these constraints by viewing $\mathbf{A}$ as allowing us to compute any dot product $\mathbf{b}_i^T\mathbf{b}_j$ in just $O(1)$ time (by reading $\mathbf{a}_{ij}$), as opposed to $O(n)$ time if we were given $\mathbf{b}_i$ and $\mathbf{b}_j$. A number of sampling approaches are known that compute a near-optimal rank-k approximation to $\mathbf{B}$ by first randomly selecting a subset of $\text{poly}(k, 1/\epsilon)$ columns that, with good probability, spans a near-optimal rank-k approximation [DV06, DMM06b, AGR16]. This approximation can then be found by computing the dot product between all other columns in $\mathbf{B}$ and the subset, requiring $O(n \cdot \text{poly}(k, 1/\epsilon))$ dot products overall – which translates to $n \cdot \text{poly}(k, 1/\epsilon)$ accesses to $\mathbf{A}$. Thus, we can compute a near-optimal low-rank approximation to $\mathbf{B}$ with a sublinear number of accesses to $\mathbf{A}$. This low-rank approximation can in turn be used to find a near-optimal low-rank approximation to $\mathbf{A}$ itself, although with a loss in approximation quality. Overall this approach yields a sublinear time algorithm for low-rank approximation of $\mathbf{A}$ accessing $n^{3/2} \cdot \text{poly}(k, 1/\epsilon)$ entries.

To achieve our final runtime of $\tilde{O}(n \cdot \text{poly}(k/\epsilon))$, we take a different but related approach, sampling the rows and columns of $\mathbf{A}$ with probabilities proportional to their ridge leverage scores. The ridge leverage scores are a natural measure of row and column importance for a broad range of linear algebraic problems, including low-rank

[1] These methods can be sped up to $\Theta(n^\omega)$ time, where $\omega < 2.373$ is the exponent of fast matrix multiplication [Wil12, LG14].

approximation [Sar06, DMM06a, AM15b, CMM17]. One can prove that sampling the rows or columns of a matrix by these scores yields a small matrix that preserves significant information about the original input. Specifically, computing a low-rank approximation of the sampled input gives a near-optimal low-rank approximation to the full matrix.

While computing these leverage scores exactly is difficult, they can be approximated efficiently via very simple iterative sampling methods, that we introduce in [CLM$^+$15, CMM17, MM17]. In particular, it is possible to approximate the ridge leverage scores of $\mathbf{B}$ with $\mathbf{B}^T\mathbf{B} = \mathbf{A}$ using just $\tilde{O}(nk)$ accesses to $\mathbf{A}$ [MM17]. We can further show that $\mathbf{B}$'s scores coarsely approximate those of $\mathbf{A}$ itself. Thus, sampling a small subset of $\mathbf{A}$'s rows and columns according to these scores yields a submatrix from which we can compute a near-optimal low-rank approximation to $\mathbf{A}$.

Our work on fast leverage score approximation algorithms initially sought to obtain sampling-based alternatives to *sparse random projections* [CW13, NN13, MM13], which give algorithms for low-rank approximation and linear regression running in $O(\text{nnz}(\mathbf{A}))$ time. We matched this state-of-the-art with sampling methods in [CLM$^+$15, CMM17]. As the result of this chapter highlights, in many cases, our sampling methods not only match, but can go significantly beyond what is achievable with random projection methods, which require $\Omega(n^2)$ time for low-rank approximation of dense PSD matrices. Beyond this example, leverage score based methods have been used in applications ranging from approximate kernel methods [MM17], to ℓ_p norm regression [CP15], to fast system solvers [KLP$^+$16, KS16], to second order optimization [ABH16, LHLS17], and matrix completion [WZZ14]. We believe that they will be even more broadly useful in future work on randomized methods for linear algebraic computation and beyond.

1.2.2 Lower Bounds for Linear Algebraic Computation

The result on PSD matrix approximation that we present in Chapter 2 is just one example of how randomization and approximation can be used to give significantly faster algorithms for basic linear algebraic problems. This algorithmic work raises the obvious question of what further progress is possible. What are the optimal runtimes for basic problems like low-rank approximation and linear regression? How important are randomization and approximation in achieving these runtimes? In Chapter 3 we attempt to understand specifically why approximation seems to be so important in designing these faster algorithms.

We begin by presenting conditional lower bounds, initially published in [MNS+18], on approximation algorithms for a number of important functions of an input matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$, including the determinant $\det(\mathbf{A})$, the trace inverse $\operatorname{tr}(\mathbf{A}^{-1})$, the trace exponential $\operatorname{tr}(\exp(\mathbf{A}))$, the Schatten p-norm $\|\mathbf{A}\|_p^p$ for any $p \neq 2$, and the matrix leverage scores. We demonstrate a tradeoff between runtime and approximation quality for all these problems.

First, we show that many of these quantities can be coarsely approximated very quickly (faster than the best known algorithms for matrix multiplication) via randomized methods. In contrast, we show that any algorithm (deterministic or randomized) for approximating the above quantities to sufficiently high accuracy can be used as a black box to give exact triangle detection in general graphs, and via a further reduction of [WW10], exact $n \times n$ Boolean matrix multiplication. This reduction implies that fast algorithms for computing these quantities to high accuracy can give fast algorithms for matrix multiplication.[2]

This result explains a recurring pattern in linear algebraic computation: if highly accurate computation is required, the only known algorithms for solving most problems are what are known as *direct methods*. Such methods include, for example, Gaussian elimination, Cholesky decomposition, and Householder orthogonalization. They typically require $\Omega(n^3)$ time. They can also typically be sped up to $\Theta(n^\omega)$ time, where $\omega < 2.373$ is the exponent of fast matrix multiplication [Wil12, LG14], but no further.[3] In contrast, fast, typically randomized, methods typically give relatively coarse approximation guarantees (i.e., they have a $\operatorname{poly}(1/\epsilon)$ runtime dependence to achieve $1 \pm \epsilon$ approximation to the above quantities). Our bounds show that this coarse approximation is in some sense required for any fast algorithm.

In the second part of Chapter 3 we present conditional lower bounds, initially published in [MW17a], which show that, given $\mathbf{A} \in \mathbb{R}^{n \times d}$, computing a near-optimal rank-k approximation to $\mathbf{A}^T\mathbf{A}$ or a number of other *kernel matrices* based off $\mathbf{A}$ cannot be done in $o(\operatorname{nnz}(\mathbf{A}) \cdot k)$[4] time without obtaining faster algorithms for general *rectangular matrix multiplication*. This lower bound matches, up to logarithmic factors, an upper bound which follows from the results of Chapter 2. Again, our conditional lower bound is a step in understanding the inherent complexity of solving

[2] Our reductions are deterministic and show that, if the the spectral sum algorithm succeeds with probability $1 - \delta$, it yields a triangle detection algorithm also succeeding with probability $1 - \delta$.

[3] See Section 1.3 for a formal definition of ω.

[4] Here and throughout, $\operatorname{nnz}(\mathbf{A})$ denotes the number of nonzero entries in $\mathbf{A}$. I.e., the time required to read all of $\mathbf{A}$ if stored in a sparse matrix format. See Section 1.3 for a general overview of linear algebraic notation.

these problems.

We conclude the chapter by discussing a number of potential directions in which to extend our results and open questions which we believe are important in establishing a more comprehensive understanding of the complexity of linear algebraic computation and the importance of randomized and approximate computation in this field.

1.2.3 Ant Colony Density Estimation

In Chapter 4 we shift to studying computation in ant colonies using the tools of randomized algorithm analysis, covering work initially published in [MSL17].

Distributed computation by social insects, including ants, termites, and bumblebees has long fascinated computer scientists. Ant colony optimization [DB11], for example, is a set of optimization techniques that employ distributed, randomized search, initial inspired by ant colony behavior. Recently, significant work in the theoretical computer science and distributed systems communities has focused on studying ant colonies from an algorithmic perspective, considering tasks such as foraging, task-allocation, and collective decision making, which are solved by these colonies [FKLS12, FK13, CDLN14, LLNR14, GMRL15, Rad17].

In this chapter we focus on the primitive of population density-estimation by ant colonies. Many ant species employ distributed population density estimation in applications ranging from quorum sensing [Pra05], to task allocation [Gor99], to appraisal of enemy colony strength [Ada90]. It has been shown that ants estimate density by tracking encounter rates – the higher the population density, the more often the ants bump into each other during a given time period [Pra05, GPT93].

We model encounter-rate-based density estimation as a simple randomized algorithm. n agents walk independently and randomly around a two-dimensional grid graph with A locations. The agents have no communication aside from the ability to detect if, in a given time step, they are located in the same position as another agent (i.e., they can detect collisions with other agents). The agents' compute the encounter rate $\frac{\#\ collisions}{t}$ at time t and use this as an estimate for the population density $d \overset{\text{def}}{=} \frac{n}{A}$.

We show that, despite dependencies inherent in the fact that nearby agents may collide repeatedly (and, worse, cannot recognize when this happens), the encounter rate gives a good estimate of the density. Specifically, after $\tilde{O}\left(\frac{\log(1/\delta)}{d\epsilon^2}\right)$, each agent will have an estimate $\tilde{d} \in (1 \pm \epsilon)d$ with probability at least $1 - \delta$. Surprisingly, this bound nearly matches what is possible if the agents are able to obtain independent samples of coin flips equal to 1 with probability d and 0 otherwise.

From a biological perspective, our work helps shed light on how ants and other social insects can obtain relatively accurate density estimates via encounter rates. It has been shown that such estimates, for example, can be used in fast consensus algorithms used for deciding between multiple candidate nests in the house-hunting process of *Temnothorax* ants [GMRL15, Rad17].

From a technical perspective, our analysis provides new tools for understanding complex dependencies in the collision probabilities of multiple random walks. We bound the strength of these dependencies using *local mixing properties* of the underlying graph. Using these properties, we extend our results beyond the two-dimensional grid to more general graphs. We show how an ant-inspired algorithm based on random walk encounter rates can be applied to estimate the number of nodes in large networks using link queries [KLS11]. We also discuss preliminary results on density estimation for robot swarms and outline future directions for research, both in studying random-walk based algorithms employed by ants, and in applying ideas from these algorithms to computational problems.

1.2.4 Computation in Spiking Neural Networks

In Chapter 5 we consider randomized computation in one of the most fascinating and difficult to understand biological systems – networks of spiking neurons like those found in the human brain. Our aim is to study basic neural primitives, such as pattern recognition, learning, and synchronization in these networks, from an algorithmic perspective. We hope to design networks solving these problems and to identify tradeoffs between runtime (convergence time of the network to an appropriately defined valid output), network complexity (e.g., the number of auxiliary neurons outside of input and output neurons used in the network), and success probability or approximation quality. We also hope to build an understanding of general network design patterns and stochastic convergence behavior. Ideally, our work can eventually be used to make biologically relevant conclusions about neural computation.

We start by defining a simple but biologically-inspired model of spiking neural networks. We model each neuron as a stochastic threshold gate, which spikes at a given discrete time t with a probability determined by the behavior of its neighbors at time $t-1$. The spiking of *excitatory* neighbors at time $t-1$ increases the neuron's membrane potential at time t, while the spiking of *inhibitory* neighbors decreases this potential. The probability of spiking at time t is given by a monotonically increasing function of this potential, e.g., by the sigmoidal function $\frac{1}{1+e^{-x}}$.

With our model in place, we consider solving the important winner-take-all (WTA) problem, in which a single output is selected from a set of firing outputs, with the non-selected outputs being suppressed. WTA resembles leader election in distributed computing and is believed to underlie many neural mechanisms, including those for visual attention [KU87, LIKB99, IK01] and competitive learning [Now89, KK94, GL09].

We present a WTA network in our basic model of computation that uses just two auxiliary inhibitor neurons to implement a competitive scheme based on the well-studied idea of lateral inhibition. We show that this network converges to a valid WTA output configuration with probability $\geq 1-\delta$ in $O(\log n \cdot \log(1/\delta))$ rounds, where n is the number of possibly firing inputs. To complement this result we prove two lower bounds. First, we show that any network with less than two auxiliary neurons cannot solve the WTA problem in any meaningful parameter regime. Specifically, such a network cannot stabilize to a valid output state for significantly longer than the time required to converge to such a state. We also show, under some additional restrictions on the network structure, that no two-inhibitor network can improve upon the convergence time achieved by our network by more than a $\log \log n$ factor. These lower bounds help clarify the dual rolls of inhibitory neurons in driving convergence to and ensuring stability of a valid configuration in fast WTA networks.

We conclude by showing that faster convergence can be achieved by employing more inhibitors and a small amount of state at each neuron used to record recent firing history. The larger number of inhibitors are used to induce multiple levels of inhibition which drive convergence at an appropriate rate depending on the number of firing outputs at a given time step.

We discuss a large number of open directions, including studying WTA for more general ‘non-binary’ input classes, using WTA within networks solving higher level computational processes, and extending our work to related symmetry breaking problems. We more broadly discuss future directions in the theoretical exploration of computation in spiking neural networks from an algorithmic perspective, including connections to the randomized algorithms for linear algebraic computation presented in the first two chapters of the thesis.

1.3 Notation and Preliminaries

In this section we lay out notation that will be used throughout the thesis. Additional notation will be defined in the specific sections where it is used.

1.3.1 General Notation

We first define a few general conventions that we will use throughout the thesis.

- For any positive integer n let $[n]$ denote the set $\{1, ..., n\}$.
- Let $f(x) = \tilde{O}(g(x))$ denote that there exists some constant c such that $f(x) = O(g(x) \cdot \log^c g(x))$.
- Let $f(x) = \text{poly}(g(x))$ denote that there exists some constant c and some degree-c polynomial p such that $f(x) = O(p(g(x)))$.

1.3.2 Matrix and Vector Notation

We next define matrix and vector notation, along with a number of basic matrix functions and properties.

Basic Notation

- Matrices and vectors are represented by bold face symbols with matrices represented by capital letters and vectors by lowercase letters, e.g., $\mathbf{M} \in \mathbb{R}^{n\times d}$, $\mathbf{x} \in \mathbb{R}^n$.
- For any $\mathbf{M} \in \mathbb{R}^{n\times d}$, let $\mathbf{m}_i \in \mathbb{R}^d$ denote its i^{th} row.
- For any $\mathbf{M} \in \mathbb{R}^{n\times d}$, let $\mathbf{M}_{i,j}$ denote the entry in the i^{th} row and j^{th} column of $\mathbf{M}$.
- For any $\mathbf{M} \in \mathbb{R}^{n\times d}$, let $\mathbf{M}^T \in \mathbb{R}^{d\times n}$ denote its transpose.
- We typically think of a vector $\mathbf{x} \in \mathbb{R}^n$ as a matrix with a single column and n rows. Let $\mathbf{x}^T \in \mathbb{R}^{1\times n}$ denote the transpose of $\mathbf{x}$, which is a matrix with a single row.
- Let $\mathbf{I}_{n\times n}$ denote the $n \times n$ identity matrix. When its dimension is clear from context we will drop the subscript.
- Let $\mathbf{0}_{n\times d}$ denote the $n \times d$ all zeros matrix. When its dimension is clear from context we will drop the subscript.
- Let $\mathbf{e}_i \in \mathbb{R}^n$ denote the standard basis vector, with a 1 at position i and zeros elsewhere.

Matrix Properties

- For any $\mathbf{M} \in \mathbb{R}^{n \times d}$, let $\text{nnz}(\mathbf{M})$ denote the *number of nonzero entries* in $\mathbf{M}$. I.e., the time required to read all of $\mathbf{M}$ if stored in a sparse matrix format. For any vector $\mathbf{x} \in \mathbb{R}^n$, let $\text{nnz}(\mathbf{x})$ similarly denote the number of nonzero entries in $\mathbf{x}$.

- For any square $\mathbf{M} \in \mathbb{R}^{n \times n}$, let $\text{tr}(\mathbf{M})$ denote its trace, the sum of $\mathbf{M}$'s diagonal entries. It is well known that, letting $\lambda_1(\mathbf{M}), ..., \lambda_n(\mathbf{M}) \in \mathbb{C}$ denote $\mathbf{M}$'s eigenvalues,
$$\text{tr}(\mathbf{M}) = \sum_{i=1}^{n} \lambda_i(\mathbf{M}).$$

- For any $\mathbf{M} \in \mathbb{R}^{n \times d}$, the Frobenius norm is given by:
$$\|\mathbf{M}\|_F \overset{\text{def}}{=} \left(\sum_{i=1}^{n} \sum_{j=1}^{d} \mathbf{M}_{i,j}^2 \right)^{1/2}.$$
Correspondingly, for $\mathbf{x} \in \mathbb{R}^n$ the ℓ_2 norm is given by $\|\mathbf{x}\|_2 \overset{\text{def}}{=} \left(\sum_{i=1}^{n} \mathbf{x}_i^2\right)^{1/2}$.

Orthonormal and Projection Matrices

- $\mathbf{Q} \in \mathbb{R}^{n \times d}$ is said to be an orthonormal matrix if it contains orthonormal columns. Equivalently, if $\mathbf{Q}^T\mathbf{Q} = \mathbf{I}_{d \times d}$.

- $\mathbf{P} \in \mathbb{R}^{n \times n}$ is an orthogonal projection matrix if and only if $\mathbf{P}^2 = \mathbf{P}$ and $\mathbf{P} = \mathbf{P}^T$. Any such matrix can be written as $\mathbf{P} = \mathbf{Q}\mathbf{Q}^T$ where $\mathbf{Q} \in \mathbb{R}^{n \times \text{rank}(\mathbf{P})}$ is an orthonormal matrix.

Singular Value Decomposition

- For any $\mathbf{M} \in \mathbb{R}^{n \times d}$, using a singular value decomposition (SVD), it is possible to write $\mathbf{M} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$, where $\mathbf{U} \in \mathbb{R}^{n \times \text{rank}(\mathbf{M})}$ and $\mathbf{V} \in \mathbb{R}^{d \times \text{rank}(\mathbf{M})}$ have orthonormal columns (the left and right singular vectors of $\mathbf{M}$), and $\mathbf{\Sigma} \in \mathbb{R}^{r \times r}$ is a positive diagonal matrix containing the singular values of $\mathbf{M}$: $\sigma_1(\mathbf{M}) \geq \sigma_2(\mathbf{M}) \geq ... \geq \sigma_{\text{rank}(\mathbf{M})}(\mathbf{M})$.

- The squared Frobenius norm (defined above) is is equal to the sum of squared singular values of $\mathbf{M}$:

$$\|\mathbf{M}\|_F^2 = \sum_{i=1}^{n}\sum_{j=1}^{d}\mathbf{M}_{i,j}^2 = \sum_{i=1}^{\text{rank}(\mathbf{M})}\sigma_i^2(\mathbf{M}).$$

- The spectral norm of $\mathbf{M}$ is defined as $\|\mathbf{M}\|_2 \stackrel{\text{def}}{=} \sigma_1(\mathbf{M})$.

- The Moore-Penrose pseudoinverse of $\mathbf{M}$ is given by $\mathbf{M}^+ = \mathbf{V}\mathbf{\Sigma}^{-1}\mathbf{U}^T$. If $\mathbf{M}$ is square and invertible $\mathbf{M}^+ = \mathbf{M}^{-1}$.

Positive Semidefinite Matrices

- A symmetric matrix $\mathbf{M} \in \mathbb{R}^{n\times n}$ is said to be positive semidefinite (PSD) if, for all $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{x}^T\mathbf{M}\mathbf{x} \geq 0$.

- If $\mathbf{M} \in \mathbb{R}^{n\times n}$, all eigenvalues $\lambda_1(\mathbf{M}) \geq \lambda_2(\mathbf{M}) \geq ... \geq \lambda_n(\mathbf{M})$ are nonnegative. Additionally, its eigenvalues are equal to its singular values: $\lambda_i(\mathbf{M}) = \sigma_i(\mathbf{M})$. Its singular value decomposition is simultaneously an eigendecomposition and can be written $\mathbf{M} = \mathbf{U}\mathbf{\Sigma}\mathbf{U}^T$.

- For any symmetric $\mathbf{M} \in \mathbb{R}^{n\times n}$ we write $\mathbf{M} \succeq \mathbf{0}$ to denote that $\mathbf{M}$ is PSD. For any symmetric $\mathbf{M}, \mathbf{N} \in \mathbb{R}^{n\times n}$ we write $\mathbf{M} \succeq \mathbf{N}$ to denote that $\mathbf{M} - \mathbf{N}$ is PSD. $\succeq$ induces a partial ordering on matrices called the *Loewner ordering.*

Fast Matrix Multiplication

- As is standard, we let ω denote the greatest value such that multiplying two $n\times n$ matrices together requires $\Omega(n^\omega)$ time in general. It is known that $2 \leq \omega < 2.373$. Further, the upper bound on ω is via the construction of an algorithm that runs in $O(n^{2.373})$ time [Wil12, LG14].

- We let $\bar{\omega}$ denote the least value for which there is a known algorithm that runs in $O(n^{\bar{\omega}})$ time. Currently $\bar{\omega} < 2.373$ [Wil12, LG14]. We will often state algorithmic runtimes in terms of $\bar{\omega}$. For example we may say that an algorithm runs in $O(nk^{\bar{\omega}-1})$ time, where $n, k \in \mathbb{Z}$ are parameters of the input. This should be interpreted as stating that, if the current best algorithm for multiplying two $n \times n$ matrices is improved (and hence $\bar{\omega}$ decreases), then the runtime of our algorithm is correspondingly improved.

Chapter 2

Sublinear Time Low-Rank Approximation of PSD Matrices

In this chapter we show how to compute a near-optimal low-rank approximation to any positive semidefinite (PSD) matrix in sublinear time. Specifically, we give a randomized algorithm that, for any PSD $\mathbf{A} \in \mathbb{R}^{n \times n}$, rank parameter $k \in \mathbb{Z}^{\geq 1}$ and accuracy parameter $\epsilon > 0$, outputs with good probability a rank-k matrix $\mathbf{B} \in \mathbb{R}^{n \times n}$, in factored form, for which

$$\|\mathbf{A} - \mathbf{B}\|_F^2 \leq (1 + \epsilon)\|\mathbf{A} - \mathbf{A}_k\|_F^2,$$

where $\mathbf{A}_k$ is the best rank-k approximation to $\mathbf{A}$ (formally defined in Section 2.1.1). The algorithm runs in $\tilde{O}(n \cdot \mathrm{poly}(k/\epsilon))$ time.[1]

When k and $1/\epsilon$ are not too large compared to the number of nonzero entries in $\mathbf{A}$, the algorithm does not need to read all entries of the matrix. Hence, this result significantly improves upon previous $\Omega(\mathrm{nnz}(\mathbf{A}))$ time algorithms based on oblivious subspace embeddings [CW13, NN13, MM13], and bypasses an $\Omega(\mathrm{nnz}(\mathbf{A}))$ time lower bound for general matrices.[2] We prove a sample complexity lower bound for low-rank approximation of PSD matrices, showing that our algorithm is close to optimal. Finally, we extend our techniques to give sublinear time algorithms for low-rank approximation of $\mathbf{A}$ in the (often stronger) spectral norm metric $\|\mathbf{A} - \mathbf{B}\|_2^2$ and for ridge regression on PSD matrices.

The results presented in this chapter were originally published in [MW17b] and

[1] For simplicity, we hide in this runtime, and throughout the introduction of this Chapter dependencies on the failure probability δ, which are never worse than $\mathrm{poly}(1/\delta)$.

[2] Recall that $\mathrm{nnz}(\mathbf{A})$ refers to the number of nonzero entries in $\mathbf{A}$ – the time required to read the full input. See Section 1.3 for an overview of notation.

represent a significant achievement in a line of work initiated in [CLM+15] and extended in [CMM17, MM17]. Our initial goal for this work was to develop simple and efficient methods for approximating the leverage scores of a matrix. This led to sampling-based algorithms matching state-of-the-art input sparsity time algorithms given by random projection methods [CW13, NN13, MM13] and iterative sampling methods combined with random projection [LMP13]. Beyond matching these methods, as highlighted in this chapter, sampling methods have allow us to push beyond prior techniques in a number of applications, including PSD matrix low-rank approximation. The interested reader may also see Christopher Musco's Ph.D. thesis [Mus18], which highlights the efficient leverage score approximation algorithms developed in [CLM+15, CMM17, MM17], along with their applications to kernel methods in machine learning.

Remark: The results of [MW17b] presented in this chapter were developed in collaboration with David Woodruff. David originally wrote up the details of the query complexity lower bound presented in Section 2.7; the presentation has been minorly modified in this thesis.

2.1 Background and Introduction to Results

Low-rank matrix approximation is a fundamental technique in numerical linear algebra. A low-rank approximation of a matrix can reveal underlying low-dimensional structure, can provide a compact way of storing a matrix in factored form, and can be quickly applied to a vector to approximate multiplication with the input matrix. Countless applications include clustering [DFK+04, FSS13, LBKW14, CEM+15], data-mining [AFK+01], information retrieval [PRTV00], learning mixtures of distributions [AM05, KSV08], recommendation systems [DKR02], topic modeling [Hof03], and web search [AFKM01, Kle99].

One of the most well-studied versions of the problem is to compute a near-optimal low-rank approximation with respect to the Frobenius norm.

Problem 2.1.1 (Near-Optimal Low-Rank Approximation)**.** *Given any* $\mathbf{A} \in \mathbb{R}^{n \times d}$*, rank parameter* $k \in \mathbb{Z}^{\geq 1}$*, and accuracy parameter* $\epsilon \geq 0$*, output* $\mathbf{M} \in \mathbb{R}^{n \times k}$*,* $\mathbf{N} \in \mathbb{R}^{d \times k}$ *such that, letting* $\mathbf{B} = \mathbf{M}\mathbf{N}^T$*,*

$$\|\mathbf{A} - \mathbf{B}\|_F^2 \leq (1+\epsilon)\|\mathbf{A} - \mathbf{A}_k\|_F^2, \tag{2.1}$$

where $\mathbf{A}_k = \arg\min_{\textit{rank-k } \mathbf{B}} \|\mathbf{A} - \mathbf{B}\|_F$.

Note that since $\mathbf{B} = \mathbf{M}\mathbf{N}^T$ where $\mathbf{M}, \mathbf{N}$ each have just k columns, $\mathbf{B}$ is rank-k. Further, any rank-k matrix can be represented in this way. Since $\mathbf{B}$ itself may have up to n^2 nonzero entries, outputting $\mathbf{B}$ directly would take $\Omega(n^2)$ time and storing $\mathbf{B}$ would take $\Omega(n^2)$ space. Thus, outputting a compact factorization $\mathbf{M}\mathbf{N}^T = \mathbf{B}$ as in Problem 2.1.1 is typically desired (and, as we will see, can be done much more efficiently, in $o(n^2)$ time).

A factorization of $\mathbf{A}_k$ can be computed exactly using the singular value decomposition – it is the projection of $\mathbf{A}$ onto the span of its top-k singular vectors. Thus, using an SVD we can solve Problem 2.1.1 deterministically, with $\epsilon = 0$. Unfortunately, computing a full SVD requires $\Omega(n^3)$ time in practice and $\Omega(n^\omega)$ time in theory, where $\omega < 2.373$ is the greatest lower bound on the exponent of fast matrix multiplication.[3]

In seminal work, Frieze, Kannan, and Vempala [FKV04] and Achlioptas and McSherry [AM07] show that using randomization and approximation, much faster runtimes are possible. Specifically, [FKV04] gives an algorithm that, assuming access to the row norms of $\mathbf{A}$, outputs rank-k $\mathbf{B}$, in factored form, such that with good probability,

$$\|\mathbf{A} - \mathbf{B}\|_F^2 \leq \|\mathbf{A} - \mathbf{A}_k\|_F^2 + \epsilon\|\mathbf{A}\|_F^2.$$

The algorithm runs in just $n \cdot \text{poly}(k/\epsilon)$ time. However $\Omega(\text{nnz}(\mathbf{A}))$ additional time is required to compute the row norms.[4] Further, the guarantee achieved can be significantly weaker than (2.1), since the error is of the form $\epsilon\|\mathbf{A}\|_F^2$ rather than $\epsilon\|\mathbf{A} - \mathbf{A}_k\|_F^2$. Note that $\|\mathbf{A} - \mathbf{A}_k\|_F^2 \ll \|\mathbf{A}\|_F^2$ exactly when $\mathbf{A}$ is well-approximated by a rank-k matrix. Related additive error algorithms with additional assumptions were given for tensors in [SWZ16].

Sarlós [Sar06] showed how to solve Problem 2.1.1 with constant probability of success using random projection in $\tilde{O}(\text{nnz}(\mathbf{A}) \cdot k/\epsilon) + n \cdot \text{poly}(k/\epsilon)$ time. This was improved by Clarkson and Woodruff [CW13] who achieved $O(\text{nnz}(\mathbf{A})) + n \cdot \text{poly}(k/\epsilon)$ time by replacing the dense random projections used by Sarlos with very sparse random projections that can be applied to $\mathbf{A}$ in $O(\text{nnz}(\mathbf{A}))$ time. See also work by Bourgain, Dirksen, and Nelson [BDN15], Cohen [Coh16], Meng and Mahoney [MM13], and Nelson and Nguyen [NN13], which further improved the degree in the $\text{poly}(k/\epsilon)$ term. For a survey, see [Woo14].

In the special case that $\mathbf{A}$ is rank-k and so $\|\mathbf{A} - \mathbf{A}_k\|_F^2 = 0$, Problem 2.1.1 is equivalent to the well studied low-rank matrix completion problem [CR09]. Much

[3] See Section 1.3 for a formal definition of ω and relatedly, the quantity $\bar{\omega}$.

[4] Recall that $\text{nnz}(\mathbf{A})$ denotes the number of nonzero entries in $\mathbf{A}$. See Section 1.3 for definitions of other linear algebraic notation.

attention has focused on completing *incoherent* low-rank matrices, whose singular directions are represented uniformly throughout the rows and columns and hence can be identified via uniform sampling and without fully accessing the matrix. Under incoherence (and often condition number) assumptions, a number of methods are able to complete a rank-k matrix in $\tilde{O}(n \cdot \text{poly}(k))$ time [JNS13, Har14]. Work by Pan et al. has also shown how to achieve such runtimes for the low-rank approximation of random matrices, which are incoherent with high probability [PLSZ17].

For general matrices, without incoherence, it is not hard to see that $\Omega(\text{nnz}(\mathbf{A}))$ is a time lower bound: if one does not read a constant fraction of entries of $\mathbf{A}$, with constant probability one can miss an entry much larger than all others, which needs to be included in the low-rank approximation.

2.1.1 Low-rank Approximation of PSD Matrices

An important class of matrices for which low-rank approximation is often applied is the set of positive semidefinite (PSD) matrices. These are symmetric matrices with all non-negative eigenvalues.[5] They arise for example as covariance matrices, graph Laplacians, Gram matrices (in particular, kernel matrices – see Chapter 3, Section 3.3 for a detailed discussion), and random dot product models [YS07]. In the multidimensional scaling literature, low-rank approximation of PSD matrices in the Frobenius norm error metric (Problem 2.1.1) corresponds to the standard 'strain minimization' problem [Dav91, CC00]. Completion of low-rank, or nearly low-rank (i.e., when $\|\mathbf{A} - \mathbf{A}_k\|_F^2 \approx 0$), PSD matrices from few entries is important in applications such as quantum state tomography [GLF$^+$10] and global positioning using local distances [SY05, YH38].

Due to its importance, a vast literature studies low-rank approximation of PSD matrices [DM05, ZTK08, KMT09, BW09, LKL10, GM13, WZ13, DLWZ14, WLZ16, TYUC16, LJS16, MM17, CW17a]. However, known algorithms either run in at least $\Omega(\text{nnz}(\mathbf{A}))$ time (which can also be achieved using sketching algorithms for general matrices), do not achieve the relative-error guarantee of Problem 2.1.1, or require strong incoherence assumptions.[6] See Table 2.1 for an overview of prior work on PSD matrix low-rank approximation.

[5] See Section 1.3 for an introduction to PSD matrices and there basic properties.

[6] Many of these algorithms satisfy the additional constraint that the low-rank approximation $\mathbf{B}$ is PSD. In the notation of Problem 2.1.1, this is equivalent to requiring that $\mathbf{M} = \mathbf{N}$. This is also now known to be possible in $O(\text{nnz}(\mathbf{A}))$ time using sketching algorithms for general matrices [CW17a].

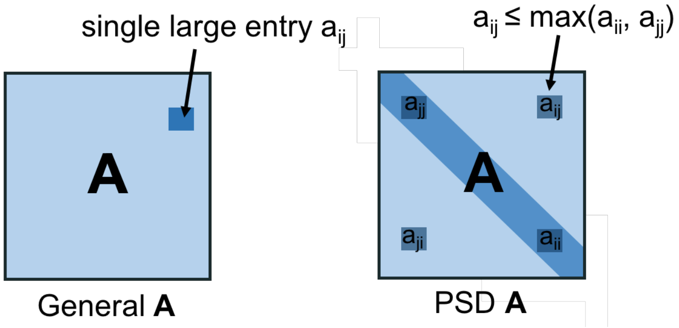

Figure 2-1: Low-rank approximation of a general matrix $\mathbf{A}$ requires $\Omega(\text{nnz}(\mathbf{A}))$ time to find a potentially randomly placed large off-diagonal entry, which must also be large in any good low-rank approximation. This lower bound does not hold for PSD $\mathbf{A}$, since each off-diagonal entry is bounded by the maximum of its two corresponding diagonal entries. So any single large entry can be found in $O(n)$ time by reading $\mathbf{A}$'s diagonal.

At the same time, the simple $\Omega(\text{nnz}(\mathbf{A}))$ time lower bound for general matrices *does not hold in the PSD case*. Positive semidefiniteness ensures that for all i, j:

$$|\mathbf{A}_{i,j}| \leq \sqrt{\mathbf{A}_{i,i} \cdot \mathbf{A}_{j,j}} \leq \max(\mathbf{A}_{i,i}, \mathbf{A}_{j,j}).$$

This is easy to see by checking that if $\mathbf{A}_{i,j} > \sqrt{\mathbf{A}_{i,i} \cdot \mathbf{A}_{j,j}}$, then $\boldsymbol{\chi}_{i,j}^T \mathbf{A} \boldsymbol{\chi}_{i,j} < 0$ where $\boldsymbol{\chi}_{i,j} \in \mathbb{R}^n$ has a 1 at position i, a -1 and position j, and zeros elsewhere. This would contradict positive semidefiniteness, which requires that $\mathbf{x}^T \mathbf{A} \mathbf{x} \geq 0$ for all $\mathbf{x} \in \mathbb{R}^n$.

So, 'hiding' a large entry in $\mathbf{A}$ requires creating a corresponding large diagonal entry. By reading the n diagonal elements, an algorithm can avoid being tricked by this approach. While far from an algorithm, this argument raises the possibility that improved, possibly sublinear, runtimes could be possible for PSD matrices. See Figure 2-1 for an illustration of the above arguement.

2.1.2 Our Contributions

We give the first sublinear time relative-error low-rank approximation algorithm for PSD matrices. Our algorithm solves Problem 2.1.1 with probability $\geq 1 - \delta$ for any $\delta > 0$. For $\delta = \Theta(1)$, it reads only $nk \cdot \text{poly}(\log n/\epsilon)$ entries of $\mathbf{A}$ and runs in $nk^{\bar{\omega}-1} \cdot \text{poly}(\log n/\epsilon)$ time, where $\bar{\omega}$ is the current smallest known exponent of matrix multiplication (Theorem 2.5.1).[7] We critically exploit the intuition discussed

[7] See Section 1.3 for a formal definition of $\bar{\omega}$, and the related quantity ω, which is the largest possible lower bound on the exponent of matrix multiplication. Currently $\bar{\omega} < 2.373$ [LG14].

in Section 2.1.1 that large off-diagonal entries cannot 'hide' in PSD matrices, but surprisingly, we require *no assumptions* on $\mathbf{A}$ aside from its positive semidefiniteness, such as incoherence or bounded condition number. As discussed, our algorithm is based on random row and column sampling using fast ridge leverage score sampling routines.

We complement our algorithm with an $\Omega(nk/\epsilon)$ time lower bound. The lower bound is information-theoretic, showing that any algorithm that reads fewer than this number of entries in the input PSD matrix cannot solve Problem 2.1.1 with probability $\geq 2/3$. Since our algorithm reads only $nk \cdot \mathrm{poly}(\log n/\epsilon)$ entries of $\mathbf{A}$, this is nearly optimal for constant ϵ. We note that the actual time complexity of our algorithm is slower by a factor of $k^{\bar{\omega}-2}$.

Finally, we show that our techniques can be extended to compute $\mathbf{M}, \mathbf{N} \in \mathbb{R}^{n\times k}$ such that $\mathbf{B} = \mathbf{M}\mathbf{N}^T$ satisfies, with good probability, the spectral norm guarantee:[8]

$$\|\mathbf{A}-\mathbf{B}\|_2^2 \leq (1+\epsilon)\|\mathbf{A}-\mathbf{A}_k\|_2^2 + \frac{\epsilon}{k}\|\mathbf{A}-\mathbf{A}_k\|_F^2. \tag{2.2}$$

Our algorithm uses just $nk^2 \cdot \mathrm{poly}(\log n/\epsilon)$ accesses to $\mathbf{A}$ and $nk^{\bar{\omega}} \cdot \mathrm{poly}(\log n/\epsilon)$ time (Theorem 2.6.3).

The guarantee of (2.2) can be stronger than (2.1) when $\|\mathbf{A}-\mathbf{A}_k\|_F^2$ is large compared to $\|\mathbf{A}-\mathbf{A}_k\|_2^2$, and is important in many applications. For example, we use this result to solve the ridge regression problem $\min_{x\in\mathbb{R}^n} \|\mathbf{A}\mathbf{x}-\mathbf{y}\|_2^2 + \lambda\|\mathbf{x}\|_2^2$ up to $(1+\epsilon)$ relative error in $\tilde{O}\left(\frac{ns_\lambda^{\bar{\omega}}}{\epsilon^{2\bar{\omega}}}\right)$ time, where $s_\lambda = \mathrm{tr}((\mathbf{A}^2+\lambda\mathbf{I})^{-1}\mathbf{A}^2)$ is the *statistical dimension* of the problem (see Theorem 2.6.7). Typically $s_\lambda \ll n$, so our runtime is sublinear and improves significantly on existing input-sparsity time results [ACW16]. For a summary of our results and comparison to prior work see, Table 2.1.

2.1.3 Algorithm Overview

We start with a general overview of our algorithmic approach.

Low-Rank Approximation via Column Subset Selection

The starting point for our approach is the fundamental fact that *any* matrix $\mathbf{A}$ contains a subset of $O(k/\epsilon)$ columns, call them $\mathbf{C}$, that span a relative-error rank-k approximation to $\mathbf{A}$ [DRVW06, DV06, DMM06b, GS12b]. Formally:

[8] Recall (see Section 1.3) that the spectral norm of any matrix $\mathbf{M} \in \mathbb{R}^{n\times d}$ is defined to be the top singular value $\|\mathbf{M}\|_2 = \sigma_1(\mathbf{M})$.

[9] Note that this bound is stated incorrectly as $\|\mathbf{A}-\mathbf{B}\|_F \leq \|\mathbf{A}-\mathbf{A}_k\|_F + \epsilon\sum_{i=1}^n(\mathbf{A}_{ii})^2$ in [DM05].

Source	Runtime	Approximation Bound
[DM05]	$nk^{\bar{\omega}-1} \cdot \text{poly}(1/\epsilon)$	$\|\mathbf{A}-\mathbf{B}\|_F \le \|\mathbf{A}-\mathbf{A}_k\|_F + \epsilon\|\mathbf{A}\|_1$[9]
[KMT09]	$nk^{\bar{\omega}-1} \cdot \text{poly}(1/\epsilon)$	$\|\mathbf{A}-\mathbf{B}\|_F \le \|\mathbf{A}-\mathbf{A}_k\|_F + \epsilon n \cdot \max_i \mathbf{A}_{ii}$
[GM13]	$\tilde{O}(n^2) + nk^{\bar{\omega}-1}\,\text{poly}(\log n/\epsilon)$	$\|\mathbf{A}-\mathbf{B}\|_F \le \|\mathbf{A}-\mathbf{A}_k\|_F + \epsilon\|\mathbf{A}-\mathbf{A}_k\|_*$
[AGR16] + [BW09]	$n\log n \cdot \text{poly}(k)$	$\|\mathbf{A}-\mathbf{B}\|_F \le (k+1)\|\mathbf{A}-\mathbf{A}_k\|_1$
[MM17]	$nk^{\bar{\omega}-1} \cdot \text{poly}(\log k/\epsilon)$	$\|\mathbf{A}^{1/2}-\mathbf{B}\|_F^2 \le (1+\epsilon)\|\mathbf{A}^{1/2}-\mathbf{A}_k^{1/2}\|_F^2$
[CW17a]	$O(\text{nnz}(\mathbf{A})) + n\,\text{poly}(k/\epsilon)$	$\|\mathbf{A}-\mathbf{B}\|_F^2 \le (1+\epsilon)\|\mathbf{A}-\mathbf{A}_k\|_F^2$
Our Results		
Thm 2.5.1	$nk^{\bar{\omega}-1} \cdot \text{poly}(\log n/\epsilon)$	$\|\mathbf{A}-\mathbf{B}\|_F^2 \le (1+\epsilon)\|\mathbf{A}-\mathbf{A}_k\|_F^2$
Thm 2.6.3	$nk^{\bar{\omega}} \cdot \text{poly}(\log n/\epsilon)$	$\|\mathbf{A}-\mathbf{B}\|_2^2 \le (1+\epsilon)\|\mathbf{A}-\mathbf{A}_k\|_2^2 + \frac{\epsilon}{k}\|\mathbf{A}-\mathbf{A}_k\|_F^2$

Table 2.1: Comparison of our results to prior work on low-rank approximation of PSD matrices. For simplicity, all stated runtimes are the required runtime to achieve the shown bound with failure probability $\delta = 1/100$. $\|\mathbf{M}\|_1 = \sum_{i=1}^n \sigma_i(\mathbf{M})$ denotes the nuclear norm of matrix $\mathbf{M}$. $\bar{\omega}$ denotes the lowest known exponent of fast matrix multiplication (see Section 1.3 for a formal definition). Currently $\bar{\omega} \approx 2.373$ [LG14]. The cited results all output $\mathbf{B}$ (in factored form), which is itself PSD. In Theorem 2.5.7 we show how to modify our algorithm to satisfy this condition, and run in $nk^{\bar{\omega}} \cdot \text{poly}(\log n/\epsilon)$ time. The table shows results that do not require incoherence assumptions on $\mathbf{A}$. For general PSD matrices, all known incoherence based results (see e.g., [Git11, GM13]) degrade to $\Omega(n^{\omega})$ runtime. Additionally, as discussed, any general low-rank approximation algorithm can be applied to PSD matrices, with state-of-the-art approaches running in input-sparsity time [CW13, MM13, NN13]. [CW17a] extends these results to the case where the output $\mathbf{B}$ is restricted to be PSD. [TYUC16] does the same but outputs $\mathbf{B}$ with rank $2k$. For a more in depth discussion of bounds obtained in prior work, see Section 2.1.4.

Theorem 2.1.2 (Theorem 1.2 of [GS12b])**.** *For any* $\mathbf{A} \in \mathbb{R}^{n \times d}$ *and any* $k \in \mathbb{Z}^{\ge 1}$*, letting* $m = \min(\lceil k/\epsilon \rceil + k - 1, d)$*, there is some* $\mathbf{C} \in \mathbb{R}^{n \times m}$ *whose columns are a subset of the columns of* $\mathbf{A}$ *such that, there exists rank-*k $\mathbf{X} \in \mathbb{R}^{d \times m}$ *with:*

$$\|\mathbf{A} - \mathbf{C}\mathbf{X}^T\|_F^2 \le (1+\epsilon)\|\mathbf{A} - \mathbf{A}_k\|_F^2. \tag{2.3}$$

Computing the best low-rank approximation to $\mathbf{A}$ using an SVD requires access to all $\Theta(n^2)$ dot products between the columns of the $\mathbf{A}$. However, given $\mathbf{C}$, just $n \cdot O(k/\epsilon)$ dot products are needed – to project the remaining columns of the matrix to the span of the subset and compute $\mathbf{X}$ satisfying (2.3).

Additionally, a subset of size $\text{poly}(k/\epsilon)$ can be identified with good probability using an intuitive approach known as *adaptive sampling* [DV06]: columns are iteratively added to the subset, with each new column being sampled with probability proportional to its norm *outside the column span* of the current subset. Formally, column $\mathbf{a}_i$ is selected with probability $\frac{\|\mathbf{a}_i - \mathbf{P_C}\mathbf{a}_i\|_2^2}{\|\mathbf{A} - \mathbf{P_C}\mathbf{A}\|_F^2}$ where $\mathbf{P_C}$ is the projection onto the current subset $\mathbf{C}$. Computing these sampling probabilities requires knowing the norm of each $\mathbf{a}_i$ along with its dot product with each column currently in $\mathbf{C}$. So, overall this approach gives a relative-error low-rank approximation using just $n \cdot \text{poly}(k/\epsilon)$ dot products between columns of $\mathbf{A}$.

The above observation is surprising – not only does every matrix contain a small column subset witnessing a near-optimal low-rank approximation, but also, such a witness can be found using significantly less information about the column span of the matrix than is required by a full SVD, via a simple random sampling approach.

Sublinear Time Low-Rank Approximation of $\mathbf{A}^{1/2}$

This fact is not immediately algorithmically useful, as computing the required dot products takes $\text{nnz}(\mathbf{A}) \cdot \text{poly}(k/\epsilon)$ time. However, given PSD $\mathbf{A}$, we can write the singular value decomposition $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{U}^T$ where $\mathbf{\Sigma}$ is a non-negative diagonal matrix of singular values, and let $\mathbf{A}^{1/2} = \mathbf{U}\mathbf{\Sigma}^{1/2}\mathbf{U}^T$ be the matrix square root of $\mathbf{A}$. Since $\mathbf{A}^{1/2}\mathbf{A}^{1/2} = \mathbf{A}$, the entry $\mathbf{A}_{i,j}$ is just the dot product between the i^{th} and j^{th} columns of $\mathbf{A}^{1/2}$. So with $\mathbf{A}$ in hand, the dot products have been 'precomputed' and the above approach yields a low-rank approximation algorithm for $\mathbf{A}^{1/2}$ running in just $n \cdot \text{poly}(k/\epsilon)$ time. Note that, aligning with our initial intuition that reading the diagonal entries of $\mathbf{A}$ is necessary to avoid the $\text{nnz}(\mathbf{A})$ time lower bound for general matrices, the diagonal entries of $\mathbf{A}$ are the column norms of $\mathbf{A}^{1/2}$, and hence their values are critical to computing the adaptive sampling probabilities.

By the above argument, given PSD $\mathbf{A}$, we can compute in $n \cdot \text{poly}(k/\epsilon)$ time a rank-k orthogonal projection matrix $\mathbf{P} \in \mathbb{R}^{n \times n}$ (in factored form) for which:

$$\|\mathbf{A}^{1/2} - \mathbf{A}^{1/2}\mathbf{P}\|_F^2 \leq (1+\epsilon)\|\mathbf{A}^{1/2} - \mathbf{A}_k^{1/2}\|_F^2.$$

This approach can be implemented using adaptive sampling [DV06], sublinear time volume sampling [AGR16], or as we show in [MM17], recursive *ridge leverage score sampling*. The ridge leverage scores are a natural interpolation between adaptive sampling and the widely studied leverage scores, which, as we will see, have a number of additional algorithmically useful properties. As discussed in [MM17], the guarantee for $\mathbf{A}^{1/2}$ is useful for a number of kernel learning methods such as kernel ridge regression. However, it is very different from the final goal of Problem 2.1.1. In fact, it is possible to show that projecting to $\mathbf{P}$ can yield an *arbitrarily bad* low-rank approximation to $\mathbf{A}$ itself (see Section 2.8).

Boosting to a True Low-Rank Approximation Using Ridge Leverage Scores

We note that, since $\mathbf{P}$ is constructed via column selection methods, it is possible to efficiently compute a factorization of $\mathbf{A}^{1/2}\mathbf{P}\mathbf{A}^{1/2}$ (see Section 2.8). Further, this matrix gives a near-optimal low-rank approximation of $\mathbf{A}$ if we use error parameter $\epsilon' = \epsilon/\sqrt{n}$ (again, see Section 2.8 for details). This approach gives a first sublinear time algorithm, but it is significantly suboptimal. Namely, it requires reading $\tilde{O}(nk/\epsilon') = \tilde{O}(n^{3/2}k/\epsilon)$ entries of $\mathbf{A}$ and takes $n^{1.69} \cdot \text{poly}(k/\epsilon)$ time using $O(n^{\bar{\omega}})$ time $n \times n$ matrix multiplication for $\bar{\omega} \approx 2.373$ [LG14].

To improve the dependence on n, we will develop a better understanding of how to select columns and rows of $\mathbf{A}$ using ridge leverage score sampling. We start by proving that the ridge leverage scores of $\mathbf{A}^{1/2}$ are within a factor of $O(\sqrt{n/k})$ of the ridge leverage scores of $\mathbf{A}$ (see Section 2.3 for a definition of these scores). By this bound, if we sample columns of $\mathbf{A}$ using approximations to the ridge leverage scores of $\mathbf{A}^{1/2}$ multiplied by $O(\sqrt{n/k})$ (these scores are computable efficiently via our algorithm developed in [MM17]), we obtain, with high probability, a sample of $\tilde{O}(\sqrt{n/k} \cdot k/\epsilon^2)$ columns that is a *projection-cost-preserving sketch* (PCP) of $\mathbf{A}$. We introduced the notion of a PCP in [CEM$^+$15]. Formally:

Definition 2.1.3 (Projection-cost-preserving sketch). *$\mathbf{C} \in \mathbb{R}^{n \times d'}$ is an (ϵ, k)-PCP of $\mathbf{A} \in \mathbb{R}^{n \times d}$ if for all rank-k projection matrices $\mathbf{P} \in \mathbb{R}^{n \times n}$:*

$$(1-\epsilon)\|\mathbf{A} - \mathbf{P}\mathbf{A}\|_F^2 \leq \|\mathbf{C} - \mathbf{P}\mathbf{C}\|_F^2 \leq (1+\epsilon)\|\mathbf{A} - \mathbf{P}\mathbf{A}\|_F^2. \tag{2.4}$$

One important property of an (ϵ, k)-PCP is that good low-rank approximations to $\mathbf{C}$ translate to good low-rank approximations of $\mathbf{A}$. More precisely, if $\mathbf{U}$ is an $n \times k$ matrix with orthonormal columns for which $\|\mathbf{C} - \mathbf{U}\mathbf{U}^T\mathbf{C}\|_F^2 \leq (1+\epsilon)\|\mathbf{C} - \mathbf{C}_k\|_F^2$, then:

$$\|\mathbf{A} - \mathbf{U}\mathbf{U}^T\mathbf{A}\|_F^2 \leq \frac{(1+\epsilon)^2}{(1-\epsilon)}\|\mathbf{A} - \mathbf{A}_k\|_F^2.$$

Letting $\mathbf{C}$ be the $n \times \tilde{O}(\sqrt{nk}/\epsilon^2)$ submatrix of $\mathbf{A}$ that we sample via ridge leverage scores, we can apply an $O(\text{nnz}(\mathbf{C}))$ time algorithm to compute a subspace $\mathbf{U} \in \mathbb{R}^{n \times k}$ whose columns span a near-optimal low-rank approximation of $\mathbf{C}$, and hence of $\mathbf{A}$ by the (ϵ, k)-PCP property. Using standard sampling techniques, we can approximately project the columns of $\mathbf{A}$ to $\mathbf{U}$, producing our final solution. This gives time complexity $O(\text{nnz}(\mathbf{C})) = n^{3/2} \cdot \text{poly}(k/\epsilon)$, improving slightly upon our first approach.

Achieving a Near-Optimal Runtime

To reduce the time to linear in n, we must further reduce the size of $\mathbf{C}$ by sampling a small subset of its rows, which themselves form a PCP of $\mathbf{C}$. To find these rows, we cannot afford to use random projection techniques, which would take at least $\Omega(\text{nnz}(\mathbf{C}))$ time, nor can we use our previous method for providing $O(\sqrt{n/k})$ overestimates to the ridge leverage scores, since $\mathbf{C}$ is no longer PSD. In fact, the row ridge leverage scores of $\mathbf{C}$ can be arbitrarily large compared to those of $\mathbf{A}^{1/2}$.

The key idea to getting around this issue is that, since $\mathbf{C}$ is a column PCP of $\mathbf{A}$, projecting its columns onto $\mathbf{A}$'s top singular vectors gives a near-optimal low-rank approximation of $\mathbf{C}$. Further, we can show that the ridge leverage scores of $\mathbf{A}^{1/2}$ (appropriately scaled) upper bound the *standard leverage scores* of this low-rank approximation. Sampling by these leverage scores is not enough to give a PCP guarantee like that of Definition 2.1.3 – they ignore the entire component of $\mathbf{C}$ not falling in the span of $\mathbf{A}$'s top singular vectors and so may significantly distort projection costs over the matrix. Further, it is unclear how to estimate even the row norms of $\mathbf{C}$, or its Frobenius norm, with $n \cdot \text{poly}(k/\epsilon)$ samples, which are necessary to implement any kind of adaptive sampling approach.

Fortunately, using that sampling at least preserves the matrix in expectation, along with a few other properties of the ridge leverage scores of $\mathbf{A}^{1/2}$, we show that, with good probability, sampling $\tilde{O}(\sqrt{nk}/\text{poly}(\epsilon))$ rows of $\mathbf{C}$ by these scores yields a subset of rows $\mathbf{R}$ satisfying a slightly weaker PCP property:

Definition 2.1.4 (Projection-cost-preserving sketch with additive error). $\mathbf{R} \in \mathbb{R}^{n' \times d}$ *is an* (ϵ, Δ, k)*-PCP of* $\mathbf{C} \in \mathbb{R}^{n \times d}$ *if for all rank-*k *projection matrices* $\mathbf{P} \in \mathbb{R}^{d \times d}$*:*

$$(1-\epsilon)\|\mathbf{C}-\mathbf{CP}\|_F^2 \leq \|\mathbf{R}-\mathbf{RP}\|_F^2 + E \leq (1+\epsilon)\|\mathbf{C}-\mathbf{CP}\|_F^2,$$

where E *is a fixed value, independent of* $\mathbf{P}$*, with* $|E| \leq \Delta \cdot \|\mathbf{C}-\mathbf{C}_k\|_F^2$ *for some* Δ*.*

We prove that $\mathbf{R}$ obtained by sampling $\tilde{O}(\sqrt{nk}/\text{poly}(\epsilon))$ rows of $\mathbf{C}$ by the ridge leverage scores of $\mathbf{A}^{1/2}$ is an $(\epsilon, O(1), k)$-PCP of $\mathbf{C}$ with good probability. Thus, since the same E error applies to all projections $\mathbf{P}$, and since it is at most a constant times the true optimum, a near-optimal low-rank approximation for $\mathbf{R}$ still translates to a near-optimal approximation for $\mathbf{C}$.

At this point $\mathbf{R}$ is a $\tilde{O}(\sqrt{nk}/\text{poly}(\epsilon)) \times \tilde{O}(\sqrt{nk}/\text{poly}(\epsilon))$ matrix, and we can run any $O(\text{nnz}(\mathbf{R}))$ time algorithm to find a good low-rank factorization to it. Specifically,

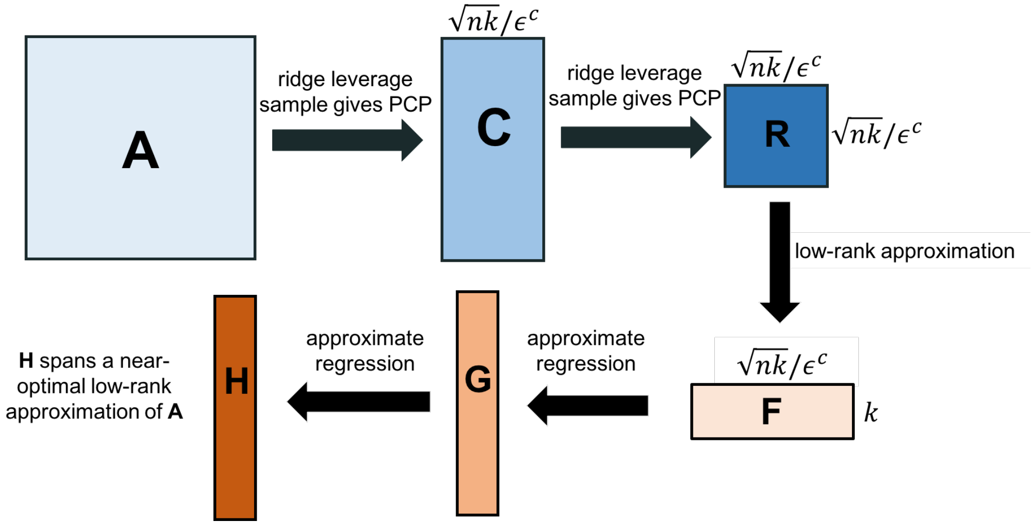

Figure 2-2: An illustration of our leverage-score-based approach. We sample $\sqrt{nk}/\operatorname{poly}(\epsilon)$ columns of $\mathbf{A}$, giving us a matrix $\mathbf{C}$. We further sample $\sqrt{nk}/\operatorname{poly}(\epsilon)$ rows to obtain a matrix $\mathbf{R}$. Due to the projection-cost-preserving sketch guarantees of ridge leverage score sampling, a near-optimal low-rank approximate to $\mathbf{R}$ can be used to give a near-optimal low-rank approximation to $\mathbf{C}$ and in turn to $\mathbf{A}$.

we can compute $\mathbf{E}, \mathbf{F} \in \mathbb{R}^{k \times \tilde{O}(\sqrt{nk}/\operatorname{poly}(\epsilon))}$ such that:

$$\|\mathbf{R} - \mathbf{E}\mathbf{F}^T\|_F^2 \leq (1+\epsilon)\|\mathbf{R} - \mathbf{R}_k\|_F^2.$$

Since $\mathbf{R}$ is a $(\epsilon, O(1), k)$-PCP for $\mathbf{C}$, by regressing the rows of $\mathbf{C}$ to the span of $\mathbf{F}$, we can obtain a near-optimal low-rank approximation to $\mathbf{C}$. We can solve this multi-response regression approximately in sublinear time via standard sampling techniques. Approximately regressing $\mathbf{A}$ to the span of this approximation using similar techniques yields our final result. The total runtime is dominated by the input-sparsity low-rank approximation of $\mathbf{R}$ requiring $O(\operatorname{nnz}(\mathbf{R})) = \tilde{O}(nk/\operatorname{poly}(\epsilon))$ time. We give an illustration of the above approach in Figure 2-2.

To improve ϵ dependencies in our final runtime, achieving sample complexity $\tilde{O}\left(\frac{nk}{\epsilon^{2.5}}\right)$, we modify this approach somewhat, showing that $\mathbf{R}$ actually satisfies a stronger *spectral norm PCP* property for $\mathbf{C}$ with good probability. This property lets us find $\mathbf{Z} \in \mathbb{R}^{d \times k}$ with $\|\mathbf{C} - \mathbf{C}\mathbf{Z}\mathbf{Z}^T\|_2^2 \leq \frac{\epsilon}{k}\|\mathbf{A} - \mathbf{A}_k\|_F^2$, from which, through a series of approximate regression steps, we can extract a low-rank approximation to $\mathbf{A}$ solving Problem 2.1.1 with good probability. This stronger spectral guarantee also lies at the core of our extensions to near-optimal spectral norm low-rank approximation (Theorem 2.6.3), ridge regression (Theorem 2.6.7), and low-rank approximation where the output is restricted to be PSD (Theorem 2.5.7).

2.1.4 Some Further Intuition on Error Guarantees

Observe that in computing a low-rank approximation of $\mathbf{A}$, we read just $\tilde{O}(n \cdot \text{poly}(k/\epsilon))$ entries of the matrix, which is, up to lower order terms, the same number of entries (corresponding to column dot products of $\mathbf{A}^{1/2}$) that we accessed to compute a low-rank approximation of $\mathbf{A}^{1/2}$ in our description above. However, these sets of entries are very different. While low-rank approximation of $\mathbf{A}^{1/2}$ looks at an $n \times \text{poly}(k/\epsilon)$ sized submatrix of $\mathbf{A}$ together with the diagonal entries, our algorithm considers a carefully chosen $\sqrt{nk}\,\text{poly}(\log n/\epsilon) \times \sqrt{nk}\,\text{poly}(\log n/\epsilon)$ submatrix together with the diagonal entries, which gives significantly more information about the spectrum of $\mathbf{A}$.

As a simple example, consider $\mathbf{A}$ with top eigenvalue $\lambda_1 = \sqrt{n}$, and $\lambda_i = 1$ for $i = 2, ...n$. $\|\mathbf{A}^{1/2}\|_F^2 = \sum_{i=1}^n \lambda_i = \sqrt{n} + n - 1$ while $\|\mathbf{A}^{1/2} - \mathbf{A}_1^{1/2}\|_F^2 = \sum_{i=2}^n \lambda_i = n - 1$. So, $\mathbf{A}^{1/2}$ has no good rank-1 approximation. Unless we set $\epsilon = O(1/\sqrt{n})$, a low-rank approximation algorithm for $\mathbf{A}^{1/2}$ can learn nothing about λ_1 and still be near-optimal. In contrast, $\|\mathbf{A}\|_F^2 = \sum_{i=1}^n \lambda_i^2 = 2n - 1$ and $\|\mathbf{A} - \mathbf{A}_1\|_F^2 = \sum_{i=2}^n \lambda_i^2 = n - 1$. So, even with $\epsilon = 1/2$, any rank-1 approximation algorithm for $\mathbf{A}$ must identify the presence of λ_1 and project this direction off the matrix. In this sense, our algorithm is able to obtain a much more accurate picture of $\mathbf{A}$'s spectrum.

With incoherence assumptions, prior work on PSD low-rank approximation [GM13] obtains the bound $\|\mathbf{A} - \mathbf{B}\|_1 \leq (1+\epsilon)\|\mathbf{A} - \mathbf{A}_k\|_1$ in sublinear time, where $\|\mathbf{M}\|_1 = \sum_{i=1}^n \sigma_i(\mathbf{M})$ is the nuclear norm of $\mathbf{M}$. Recent work ([AGR16] in combination with [BW09]) gives $\|\mathbf{A} - \mathbf{B}\|_F \leq (k+1)\|\mathbf{A} - \mathbf{A}_k\|_1$ without the incoherence assumption. These nuclear norm bounds are closely related to approximation bounds for $\mathbf{A}^{1/2}$ and it is not hard to see that neither require λ_1 to be detected in the example above, and so in this sense are weaker than our Frobenius norm bound.

A natural question if even stronger bounds are possible. Can we compute $\mathbf{B}$ with:

$$\|\mathbf{A} - \mathbf{B}\|_2^2 \leq (1+\epsilon)\|\mathbf{A} - \mathbf{A}_k\|_2^2 \tag{2.5}$$

in sublinear time? We partially answer this question in Theorem 2.6.3. We show that in $\tilde{O}(nk^{\bar{\omega}}\,\text{poly}(\log n/\epsilon))$ time, we can find $\mathbf{B}$, in factored form, satisfying (2.2),

$$\|\mathbf{A} - \mathbf{B}\|_2^2 \leq (1+\epsilon)\|\mathbf{A} - \mathbf{A}_k\|_2^2 + \frac{\epsilon}{k}\|\mathbf{A} - \mathbf{A}_k\|_F^2$$

with good probability. Significantly improving the above bound seems difficult: it is easy to see that computing $\mathbf{B}$ (in factored form) that satisfies the relative error spectral norm guarantee of (2.5) requires $\Omega(n^2)$ time. Consider a random input $\mathbf{A}$ that is equal to the identity except with $\mathbf{A}_{i,j} = \mathbf{A}_{j,i} = 1$ for some uniform random pair (i, j). Finding (i, j) requires $\Omega(n^2)$ queries to $\mathbf{A}$. However, it is necessary to achieve a relative error spectral norm guarantee with $\epsilon < 3$ since $\|\mathbf{A}\|_2^2 = 4$ while $\|\mathbf{A} - \mathbf{A}_1\|_2^2 = 1$ where $\mathbf{A}_1$ is all zeros with ones at its (i, i), (j, j), (i, j), and (j, i) entries.

A similar argument shows that relative error low-rank approximation in higher Schatten-p norms, i.e., $\|\mathbf{A} - \mathbf{B}\|_p^p$ for $p > 2$ requires superlinear dependence on n (where $\|\mathbf{M}\|_p^p = \sum_{i=1}^n \sigma_i^p(\mathbf{M})$.) We can set $\mathbf{A}$ to be the identity but with an all ones block on a uniform random subset of $n^{1/p}$ indices. This block has associated eigenvalue $\lambda_1 = n^{1/p}$ and so, since all other $(n - n^{1/p})$ eigenvalues of $\mathbf{A}$ are 1, $\|\mathbf{A}\|_p^p = \Theta(n)$, and the block must be recovered to give a relative error approximation to $\|\mathbf{A} - \mathbf{A}_1\|_p^p$. However, as the block is placed uniformly at random and contains just $n^{2/p}$ entries, finding even a single entry requires $\Omega(n^{2-2/p})$ queries to $\mathbf{A}$ – superlinear for $p > 2$.

2.1.5 Road Map

We present our results as follows. Aside from some of the background results presented in Section 2.2, all results were originally published in [MW17b].

Section 2.2: Ridge Leverage Scores. We introduce the ridge leverage scores, our prior results on approximating them quickly [CMM17, MM17], and approximation guarantees that hold for ridge leverage score sampling.

Section 2.3: Column Sampling. We show that the ridge leverage scores of $\mathbf{A}$ are within an $O(\sqrt{n/k})$ factor of those of $\mathbf{A}^{1/2}$, letting us use the fast ridge leverage score sampling algorithm of [MM17] to sample $\mathbf{C}$, a subset of $\tilde{O}(\sqrt{nk}/\epsilon^2)$ columns of $\mathbf{A}$ that form a PCP of the matrix.

Section 2.4: Row Sampling. We discuss how to further accelerate our algorithm by sampling rows from $\mathbf{C}$ to obtain a PCP, letting us to achieve linear in n runtime.

Section 2.5: Full Algorithm. We use the primitives in the previous sections along with standard approximate regression techniques to give our full sublinear time low-rank approximation algorithm.

Section 2.6: Spectral Norm Bounds. We modify the algorithm of Section 2.5 to give the tighter spectral norm approximation of (2.2) and discuss applications to sublinear time ridge regression.

Section 2.7: Lower Bounds. We show that our algorithm is nearly optimal – any relative error low-rank approximation algorithm must read $\Omega(nk/\epsilon)$ entries of $\mathbf{A}$.

Section 2.8: Simple Sublinear Time Algorithm. We demonstrate that a low-rank approximation for $\mathbf{A}^{1/2}$ does not directly yield one for $\mathbf{A}$, but that a simple approach can be used to convert a low-rank approximation to $\mathbf{A}^{1/2}$ to a near-optimal one for $\mathbf{A}$ in $n^{1.69} \cdot \text{poly}(k/\epsilon)$ time.

2.2 Ridge Leverage Score Sampling

Our main algorithmic tool will be ridge leverage score sampling, which is used to identify a small subset of columns of $\mathbf{A}$ that span a good low-rank approximation of the matrix. In this section we give a general introduction to and some basic results on the ridge leverage scores, drawing on our work in [CMM17, MM17, MW17b]. The results here constitute a general (but very incomplete) overview of sampling methods for randomized linear algebra, useful beyond just the work presented in this chapter.

In Section 2.2.1 we define the ridge leverage scores and the related standard leverage scores and give a few basic properties of these scores that we use throughout our analysis. In Section 2.2.2 we prove that sampling the rows or columns of a matrix by their (ridge) leverage scores yield, with good probability, an approximation to the matrix that can be used, e.g., to approximately solve fundamental low-rank approximation and regression problems. Finally, in Section 2.2.3 we discuss efficient computation of these scores.

2.2.1 Leverage Score Definitions and Basic Properties

We first introduce the ridge leverage scores and some of their basic properties. Beyond low-rank approximation, the ridge leverage scores have been used in algorithms for a number of problems including ordinary least squares regression [LMP13], ridge regression [AM15a], and graph sparsification [KLM+17]. In the general form they are given by:

Definition 2.2.1 (Ridge Leverage Scores). *For any $\mathbf{A} \in \mathbb{R}^{n\times d}$ and ridge parameter $\lambda \geq 0$, letting $\mathbf{a}_i \in \mathbb{R}^n$ be the i^{th} column of $\mathbf{A}$, the i^{th} λ column ridge leverage score of $\mathbf{A}$ is:*

$$\tau_{i,\lambda}(\mathbf{A}) = \mathbf{a}_i^T \left(\mathbf{A}\mathbf{A}^T + \lambda\mathbf{I}\right)^+ \mathbf{a}_i.$$

Recall that for any matrix $\mathbf{M}$, $\mathbf{M}^+$ denotes its Moore-Penrose pseudoinverse (see Section 1.3). Unless $\lambda = 0$ and $\mathbf{A}$ is singular, all of $(\mathbf{A}\mathbf{A}^T + \lambda\mathbf{I})$'s eigenvalues are positive and $(\mathbf{A}\mathbf{A}^T + \lambda\mathbf{I})^+ = (\mathbf{A}\mathbf{A}^T + \lambda\mathbf{I})^{-1}$. Note that we define the ridge leverage scores for $\mathbf{A}$'s columns. However, ridge leverages scores can be defined for $\mathbf{A}$'s rows as well: as the corresponding column ridge leverage scores of $\mathbf{A}^T$.

For low-rank approximation, we set the ridge parameter λ appropriately, giving us the rank-k ridge leverage scores:

Definition 2.2.2 (Rank-k Ridge Leverage Scores)**.** *For any $\mathbf{A} \in \mathbb{R}^{n\times d}$, letting $\mathbf{a}_i \in \mathbb{R}^n$ be the i^{th} column of $\mathbf{A}$, the i^{th} rank-k column ridge leverage score of $\mathbf{A}$ is given by $\tau_i^k(\mathbf{A}) = \tau_{i,\lambda}(\mathbf{A})$ (Definition 2.2.1) with $\lambda = \frac{\|\mathbf{A}-\mathbf{A}_k\|_F^2}{k}$.*

Above $\mathbf{I}$ is the $n \times n$ identity matrix. It is not hard to check that $0 < \tau_i^k(\mathbf{A}) < 1$ for all i. Since we use these scores as sampling probabilities, it is critical that the sum of scores, and hence the size of the subsets we sample, is not too large. We have the following bound:

Lemma 2.2.3 (Sum of Ridge Leverage Scores)**.** *For any $\mathbf{A} \in \mathbb{R}^{n\times d}$ and any k,*

$$\sum_{i=1}^{d} \tau_i^k(\mathbf{A}) \leq 2k.$$

Proof. We rewrite Definition 2.2.2 using $\mathbf{A}$'s singular value decomposition $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$.

$$\begin{aligned}\tau_i(\mathbf{A}) &= \mathbf{a}_i^T\left(\mathbf{U}\mathbf{\Sigma}^2\mathbf{U}^T + \frac{\|\mathbf{A}-\mathbf{A}_k\|_F^2}{k}\mathbf{U}\mathbf{U}^T\right)^{-1}\mathbf{a}_i\\ &= \mathbf{a}_i^T\left(\mathbf{U}\bar{\mathbf{\Sigma}}\mathbf{U}^T\right)\mathbf{a}_i,\end{aligned}$$

where $\bar{\mathbf{\Sigma}}$ is the diagonal matrix with $\bar{\mathbf{\Sigma}}_{i,i} = \frac{1}{\sigma_i^2(\mathbf{A}) + \frac{\|\mathbf{A}-\mathbf{A}_k\|_F^2}{k}}$. We then have:

$$\sum_{i=1}^{n} \tau_i(\mathbf{A}) = \operatorname{tr}\left(\mathbf{A}^T\mathbf{U}\bar{\Sigma}\mathbf{U}^T\mathbf{A}\right) = \operatorname{tr}\left(\mathbf{V}\mathbf{\Sigma}\bar{\mathbf{\Sigma}}\mathbf{\Sigma}\mathbf{V}^T\right) = \operatorname{tr}(\mathbf{\Sigma}^2\bar{\mathbf{\Sigma}}). \tag{2.6}$$

$\mathbf{\Sigma}$ and $\bar{\mathbf{\Sigma}}$ are diagonal and we can compute:

$$(\mathbf{\Sigma}^2\bar{\mathbf{\Sigma}})_{i,i} = \frac{\sigma_i^2(\mathbf{A})}{\sigma_i^2(\mathbf{A}) + \frac{\|\mathbf{A}-\mathbf{A}_k\|_F^2}{k}}. \tag{2.7}$$

For $i \leq k$ we simply upper bound (2.7) by 1. Combined with (2.6) we have:

$$\begin{aligned}\sum_{i=1}^{n} \tau_i(\mathbf{A}) &= \operatorname{tr}(\mathbf{\Sigma}^2 \bar{\mathbf{\Sigma}}) \\ &\leq k + \sum_{i=k+1}^{n} \frac{\sigma_i^2(\mathbf{A})}{\sigma_i^2(\mathbf{A}) + \frac{\|\mathbf{A}-\mathbf{A}_k\|_F^2}{k}} \\ &\leq k + k \sum_{i=k+1}^{n} \frac{\sigma_i^2(A)}{\|\mathbf{A}-\mathbf{A}_k\|_F^2} \\ &= 2k.\end{aligned}$$

□

The ridge leverage scores are a generalization of the standard leverage scores of $\mathbf{A}$, which are defined as:

Definition 2.2.4 (Standard Leverage Scores – see e.g., [Woo14])**.** *For any $\mathbf{A} \in \mathbb{R}^{n \times d}$, letting $\mathbf{a}_i \in \mathbb{R}^n$ be the i^{th} column of $\mathbf{A}$, the i^{th}* column leverage score *of $\mathbf{A}$ is:*

$$\ell_i(\mathbf{A}) = \mathbf{a}_i^T \left(\mathbf{A}\mathbf{A}^T\right)^+ \mathbf{a}_i.$$

By writing $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ in its SVD, one can see that standard leverage scores are just the squared column norms of $\mathbf{V}^T$, and sum to $\operatorname{rank}(\mathbf{A})$, which for $\mathbf{A} \in \mathbb{R}^{n \times d}$ may be as large as $\min(n, d)$. Sampling columns with probabilities proportional to their leverage scores yields what is known as a spectral approximation to $\mathbf{A}$ – formally defined in Lemma 2.2.6, via a matrix Chernoff bound.

In the ridge leverage scores, the addition of the weighted identity (or 'ridge') $\lambda\mathbf{I}$, 'dampens' contributions from smaller singular directions of $\mathbf{A}$, decreasing the sum of the scores and allowing us to sample fewer columns. At the same time, it introduces error dependent on the size of λ. In Definition 2.80 we set $\lambda = \frac{\|\mathbf{A}-\mathbf{A}_k\|_F^2}{k}$, which ultimately gives an approximation to $\mathbf{A}$ from which it is possible to output a near-optimal low-rank approximation to the original matrix.

Before giving the above mentioned sampling results, we formally show that the standard leverage scores of $\mathbf{A}$ are upper bounded by the row norms of any orthonormal matrix spanning $\mathbf{A}$'s rows. If the span of the orthogonal matrix is equal to the span of $\mathbf{A}$'s rows, this inequality becomes an equality.

Lemma 2.2.5. *For any $\mathbf{A} \in \mathbb{R}^{n \times d}$, let $\mathbf{Z} \in \mathbb{R}^{d \times d'}$ have orthonormal columns that span the rows of $\mathbf{A}$. Then, for all $i \in [d]$, the i^{th} column leverage score of $\mathbf{A}$ (Definition*

2.2.4) is bounded by:

$$\ell_i(\mathbf{A}) \leq \|\mathbf{z}_i\|_2^2,$$

where $\mathbf{z}_i$ is the i^{th} row of $\mathbf{Z}$. If the column span of $\mathbf{Z}$ is equal to the row span of $\mathbf{A}$ then $\ell_i(\mathbf{A}) = \|\mathbf{z}_i\|_2^2$.

Proof. Write the singular value decomposition $\mathbf{A} = \mathbf{U\Sigma V}^T$ and recall that $\mathbf{e}_i$ denotes the i^{th} standard basis vector, with a 1 at position i and zeros elsewhere. We have:

$$\begin{aligned}
\ell_i(\mathbf{A}) &= \mathbf{e}_i^T \mathbf{A}^T(\mathbf{A}\mathbf{A}^T)^+\mathbf{A}\mathbf{e}_i \\
&= \mathbf{e}_i^T \mathbf{V\Sigma U}^T(\mathbf{U\Sigma}^{-2}\mathbf{U}^T)\mathbf{U\Sigma V}^T\mathbf{e}_i \\
&= \mathbf{e}_i^T \mathbf{VIV}^T\mathbf{e}_i \\
&= \|\mathbf{v}_i\|_2^2. \qquad (2.8)
\end{aligned}$$

where $\mathbf{v}_i$ is the i^{th} row of $\mathbf{V}$. Now, the columns of $\mathbf{V}$ fall within the row span of $\mathbf{A}$ and so are spanned by the columns of $\mathbf{Z}$. So we can write $\mathbf{V} = \mathbf{ZZ}^T\mathbf{V}$, which gives:

$$\begin{aligned}
\|\mathbf{v}_i\|_2^2 = \mathbf{e}_i^T\mathbf{VV}^T\mathbf{e}_i &= \mathbf{e}_i\mathbf{ZZ}^T\mathbf{VV}^T\mathbf{ZZ}^T\mathbf{e}_i \\
&\leq \mathbf{e}_i\mathbf{ZZ}^T\mathbf{ZZ}^T\mathbf{e}_i \\
&= \mathbf{e}_i\mathbf{ZIZ}^T\mathbf{e}_i \\
&= \|\mathbf{z}_i\|_2^2.
\end{aligned}$$

Combined with (2.8) this gives $\ell_i(\mathbf{A}) \leq \|\mathbf{z}_i\|_2^2$, and hence the lemma. If the column span of $\mathbf{Z}$ is equal to the row span of $\mathbf{A}$, then above we in fact have $\|\mathbf{v}_i\|_2^2 = \|\mathbf{z}_i\|_2^2$, giving $\ell_i(\mathbf{A}) = \|\mathbf{z}_i\|_2^2$ in this case. □

2.2.2 Approximation Bounds

We now prove basic matrix approximation results that hold when sampling rows or columns using both the standard leverage scores and ridge leverage scores. We start by proving general *spectral approximation bounds* that directly bound the degree to which a matrix is approximated after sampling. We then prove more specialized approximation bounds, which show that after leverage score sampling, the sampled matrix can be used in place of the original matrix to approximately solve low-rank approximation and regression problems.

Spectral Approximation Bounds

We first prove what are known as spectral approximation bounds, or subspace embedding bounds (see, e.g. [Sar06, SS08, CW17b]). These bounds show that, after sampling the columns of a matrix $\mathbf{A}$ using probabilities proportional to their leverage scores, the norm of $\mathbf{x}^T\mathbf{A}$ is approximately preserved for all $\mathbf{x} \in \mathbb{R}^n$. The more columns sampled, the higher accuracy the approximation will be.

We first show a spectral approximation bound for standard leverage score sampling, via the application of a matrix Chernoff bound [Tro15]. This type of bound is standard – see e.g. [DMM06a, LMP13, CLM$^+$15].

Lemma 2.2.6 (Leverage Score Sampling Spectral Approximation)**.** *For any* $\mathbf{A} \in \mathbb{R}^{n\times d}$ *and* $i \in [d]$*, let* $\tilde{\ell}_i \geq \ell_i(\mathbf{A})$ *be an overestimate for* $\mathbf{A}$*'s* i^{th} *column leverage score (Definition 2.2.4. Let* $p_i = \frac{\tilde{\ell}_i}{\sum_i \tilde{\ell}_i}$ *and for any* $\epsilon, \delta \in (0, 1/2]$*,* $t \geq \frac{32\log(\text{rank}(\mathbf{A})/\delta)}{\epsilon^2}\sum_i \tilde{\ell}_i$*. Construct* $\mathbf{C}$ *by sampling* t *columns of* $\mathbf{A}$*, each independently set to* $\frac{1}{\sqrt{tp_i}}\mathbf{a}_i$ *with probability* p_i*. Then, with probability* $\geq 1-\delta$*:*

$$(1-\epsilon)\mathbf{C}\mathbf{C}^T \preceq \mathbf{A}\mathbf{A}^T \preceq (1+\epsilon)\mathbf{C}\mathbf{C}^T. \tag{2.9}$$

Recall that for $\mathbf{M}, \mathbf{N} \in \mathbb{R}^{n\times n}$, $\mathbf{M} \succeq \mathbf{N}$ indicates that $\mathbf{M} - \mathbf{N}$ is PSD (see Section 1.3.) Thus, (2.9) implies that, for any $\mathbf{x} \in \mathbb{R}^n$:

$$\begin{aligned}\mathbf{x}^T(\mathbf{A}\mathbf{A}^T - (1-\epsilon)\mathbf{C}\mathbf{C}^T)\mathbf{x} \geq 0 &\\ \mathbf{x}^T\mathbf{A}\mathbf{A}^T\mathbf{x} &\geq (1-\epsilon)\mathbf{x}^T\mathbf{C}\mathbf{C}^T\mathbf{x}\\ \|\mathbf{x}^T\mathbf{A}\|_2^2 &\geq (1-\epsilon)\|\mathbf{x}^T\mathbf{C}\|_2^2.\end{aligned}$$

Similarly, it gives $(1+\epsilon)\|\mathbf{x}^T\mathbf{C}\|_2^2 \geq \|\mathbf{x}^T\mathbf{A}\|_2^2$. Thus, we can interpret (2.9) as showing that $\mathbf{C}$ can be used to approximate the norm of $\mathbf{x}^T\mathbf{A}$ up to $(1 \pm \epsilon)$ error, for any $\mathbf{x} \in \mathbb{R}^n$.

We prove Lemma 2.2.6 using standard techniques for showing matrix concentration results.

Proof of Lemma 2.2.6. Write the singular value decomposition $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$. By (2.8) we have $\ell_i(\mathbf{A}) = \|\mathbf{v}_i\|_2^2$ where $\mathbf{v}_i$ is the i^{th} row of $\mathbf{V}$.

Let $\mathbf{Y} = \mathbf{\Sigma}^{-1}\mathbf{U}^T\left(\mathbf{C}\mathbf{C}^T - \mathbf{A}\mathbf{A}^T\right)\mathbf{U}\mathbf{\Sigma}^{-1}$ and for $j \in [t]$, define a set of independent and identically distributed random variables $\mathbf{X}_1, ..., \mathbf{X}_t$ by:

$$\mathbf{X}_j = \frac{1}{t}\cdot\mathbf{\Sigma}^{-1}\mathbf{U}^T\left(\frac{1}{p_i}\mathbf{a}_i\mathbf{a}_i^T - \mathbf{A}\mathbf{A}^T\right)\mathbf{U}\mathbf{\Sigma}^{-1} \text{ with probability } p_i.$$

We can rewrite the random variable $\mathbf{Y}$ as:

$$\mathbf{Y} = \sum_{j=1}^{t} \left[\mathbf{\Sigma}^{-1}\mathbf{U}^T \left(\mathbf{c}_j\mathbf{c}_j^T - \frac{1}{t}\mathbf{A}\mathbf{A}^T \right) \mathbf{U}\mathbf{\Sigma}^{-1} \right] \overset{\text{def}}{=} \sum_{j=1}^{t} [\mathbf{X}_j] .$$

We have $\mathbb{E}[\mathbf{X}_j] = \sum_{i=1}^{d} p_i \left[\frac{1}{p_i}\mathbf{a}_i\mathbf{a}_i^T - \mathbf{A}\mathbf{A}^T \right] = \mathbf{0}$, which gives $\mathbb{E}[\mathbf{Y}] = \mathbf{0}$. We have the following claim:

Claim 2.2.7. *If $\|\mathbf{Y}\|_2 \leq \epsilon/2$ then $(1-\epsilon)\mathbf{C}\mathbf{C}^T \preceq \mathbf{A}\mathbf{A}^T \preceq (1+\epsilon)\mathbf{C}\mathbf{C}^T$.*

Proof. $\|\mathbf{Y}\|_2 \leq \epsilon/2$ can be equivalently written as

$$-\frac{\epsilon}{2}\mathbf{I} \preceq \mathbf{Y} \preceq \frac{\epsilon}{2}\mathbf{I}.$$

Writing $\mathbf{C}\mathbf{C}^T - \mathbf{A}\mathbf{A}^T = \mathbf{U}\mathbf{\Sigma}\mathbf{Y}\mathbf{\Sigma}\mathbf{U}$ we can thus bound:

$$-\frac{\epsilon}{2}\mathbf{U}\mathbf{\Sigma}\mathbf{I}\mathbf{\Sigma}\mathbf{U} \preceq \mathbf{C}\mathbf{C}^T - \mathbf{A}\mathbf{A}^T \preceq \frac{\epsilon}{2}\mathbf{U}\mathbf{\Sigma}\mathbf{I}\mathbf{\Sigma}\mathbf{U}.$$

Noting that $\mathbf{U}\mathbf{\Sigma}\mathbf{I}\mathbf{\Sigma}\mathbf{U} = \mathbf{A}\mathbf{A}^T$ we thus have:

$$(1-\epsilon/2)\mathbf{A}\mathbf{A}^T \preceq \mathbf{C}\mathbf{C}^T \preceq (1+\epsilon/2)\mathbf{A}\mathbf{A}^T$$

which gives

$$\frac{1}{1+\epsilon/2}\mathbf{C}\mathbf{C}^T \preceq \mathbf{A}\mathbf{A}^T \preceq \frac{1}{1-\epsilon/2}\mathbf{C}\mathbf{C}^T.$$

The claim follows since for any $\epsilon \in (0,1]$, $\frac{1}{1+\epsilon/2} \geq 1-\epsilon$ and $\frac{1}{1-\epsilon/2} \leq 1+\epsilon$. □

By Claim 2.2.7, to prove Lemma 2.2.6, it suffices to show that $\|\mathbf{Y}\|_2 \leq \epsilon/2$ with probability $\geq 1-\delta$. We prove this bound using a matrix Bernstein inequality from [Tro15]. Applying this bound requires upper bounds on the spectral norm of each $\mathbf{X}_j$ and on the variance of $\mathbf{Y}$. We first prove a key property of the leverage scores:

Lemma 2.2.8. *For any $\mathbf{A} \in \mathbb{R}^{n\times d}$ and any $i \in [n]$, $\frac{1}{\ell_i(\mathbf{A})} \cdot \mathbf{a}_i\mathbf{a}_i^T \preceq \mathbf{A}\mathbf{A}^T$.*

Proof. Let $(\mathbf{A}\mathbf{A}^T)^{+/2} = \mathbf{U}\mathbf{\Sigma}^{-1}\mathbf{U}^T$. Any vector $\mathbf{x} \in \mathbb{R}^d$ that is in $\mathbf{A}$'s column span can be written as $\mathbf{A}\mathbf{z}$ for some $\mathbf{z} \in \mathbb{R}^d$ in $\mathbf{A}$'s row span, which is equal to the row span of $\mathbf{V}^T$. We can thus write $\mathbf{x} = \mathbf{A}\mathbf{V}\mathbf{w} = \mathbf{U}\mathbf{\Sigma}$ for some $\mathbf{w} \in \mathbb{R}^{\text{rank}(\mathbf{A})}$. Setting $\mathbf{y} = \mathbf{U}\mathbf{\Sigma}^2$

we can thus write:

$$\mathbf{x} = (\mathbf{A}\mathbf{A}^T)^{+/2}\mathbf{y} = \mathbf{U}\boldsymbol{\Sigma}^{-1}\mathbf{U}^T\mathbf{U}\boldsymbol{\Sigma}^2\mathbf{w} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{w}. \tag{2.10}$$

Using (2.10) we can write for any $\mathbf{x}$ in $\mathbf{A}$'s column span:

$$\begin{aligned}\mathbf{x}^T\left(\mathbf{a}_i\mathbf{a}_i^T\right)\mathbf{x} &= \mathbf{y}^T(\mathbf{A}\mathbf{A}^T)^{+/2}\mathbf{a}_i\mathbf{a}_i^T(\mathbf{A}\mathbf{A}^T)^{+/2}\mathbf{y}\\ &\leq \ell_i(\mathbf{A})\|\mathbf{y}\|_2^2 \end{aligned} \tag{2.11}$$

since $(\mathbf{A}\mathbf{A}^T)^{+/2}\mathbf{a}_i\mathbf{a}_i^T(\mathbf{A}\mathbf{A}^T)^{+/2}$ is rank-1 and so has maximum eigenvalue

$$\operatorname{tr}\left((\mathbf{A}\mathbf{A}^T)^{+/2}\mathbf{a}_i\mathbf{a}_i^T(\mathbf{A}\mathbf{A}^T)^{+/2}\right) = \mathbf{a}_i^T(\mathbf{A}\mathbf{A}^T)^{+}\mathbf{a}_i = \ell_i(\mathbf{A})$$

by the cyclic property of trace. Further,

$$\mathbf{x}^T\mathbf{A}\mathbf{A}^T\mathbf{x} = \mathbf{y}^T(\mathbf{A}\mathbf{A}^T)^{+/2}\mathbf{A}\mathbf{A}^T(\mathbf{A}\mathbf{A}^T)^{+/2}\mathbf{y} = \|\mathbf{y}\|_2^2.$$

Combined with (2.11) this gives us, for any $\mathbf{x}$ in the column span of $\mathbf{A}$:

$$\frac{1}{\ell_i(\mathbf{A})}\cdot\mathbf{x}^T\left(\mathbf{a}_i\mathbf{a}_i^T\right)\mathbf{x} \leq \mathbf{x}^T\mathbf{A}\mathbf{A}^T\mathbf{x}. \tag{2.12}$$

For any $\mathbf{x}\in\mathbb{R}^n$ not in the column span of $\mathbf{A}$, we can write $\mathbf{x} = \mathbf{P}\mathbf{x} + (\mathbf{I}-\mathbf{P})\mathbf{x}$ where $\mathbf{P}$ is the projection to this span. We then have:

$$\mathbf{x}^T\left(\mathbf{a}_i\mathbf{a}_i^T\right)\mathbf{x} = \mathbf{x}^T\mathbf{P}\left(\mathbf{a}_i\mathbf{a}_i^T\right)\mathbf{P}\mathbf{x} + 0$$

and

$$\mathbf{x}^T\mathbf{A}\mathbf{A}^T\mathbf{x} = \mathbf{x}^T\mathbf{P}\mathbf{A}\mathbf{A}^T\mathbf{P}\mathbf{x} + 0.$$

The lemma follows by applying (2.12) to $\mathbf{P}\mathbf{x}$, which is in the column span of $\mathbf{A}$. □

Using Lemma 2.2.8 we can prove:

Claim 2.2.9. *For all $j\in[t]$, $\|\mathbf{X}_j\|_2 \leq \frac{\epsilon^2}{32\log(\operatorname{rank}(\mathbf{A})/\delta)}$ with probability* 1.

Proof. Lemma 2.2.8 gives:

$$\frac{1}{\ell_i(\mathbf{A})}\cdot\boldsymbol{\Sigma}^{-1}\mathbf{U}^T\mathbf{a}_i\mathbf{a}_i^T\mathbf{U}\boldsymbol{\Sigma}^{-1} \preceq \boldsymbol{\Sigma}^{-1}\mathbf{U}^T\mathbf{A}\mathbf{A}^T\mathbf{U}\boldsymbol{\Sigma}^{-1} = \mathbf{I}.$$

And so, using the assumption that for all i, $\tilde{\ell}_i \geq \ell_i(\mathbf{A})$ and that $t \geq \frac{32\log(\text{rank}(\mathbf{A})/\delta)}{\epsilon^2}$:

$$\begin{aligned}\mathbf{0} \preceq \frac{1}{tp_i}\mathbf{\Sigma}^{-1}\mathbf{U}^T\mathbf{a}_i\mathbf{a}_i^T\mathbf{U}\mathbf{\Sigma}^{-1} &\preceq \frac{\epsilon^2}{32\log(\text{rank}(\mathbf{A})/\delta)\tilde{\ell}_i}\mathbf{\Sigma}^{-1}\mathbf{U}^T\mathbf{a}_i\mathbf{a}_i^T\mathbf{U}\mathbf{\Sigma}^{-1}\\ &\preceq \frac{\epsilon^2}{32\log(\text{rank}(\mathbf{A})/\delta)}\mathbf{I}. \end{aligned} \tag{2.13}$$

Additionally,

$$\mathbf{0} \preceq \frac{1}{t}\mathbf{\Sigma}^{-1}\mathbf{U}^T\mathbf{A}\mathbf{A}^T\mathbf{U}\mathbf{\Sigma}^{-1} \preceq \frac{\epsilon^2}{32\log(\text{rank}(\mathbf{A})/\delta)}\mathbf{I}, \tag{2.14}$$

using that $\sum_i \tilde{\ell}_i \geq 1$ since $\sum_i \tilde{\ell}_i \geq \sum_i \ell_i(\mathbf{A}) = \text{rank}(\mathbf{A})$. Recalling that

$$\mathbf{X}_j = \frac{1}{t}\cdot\mathbf{\Sigma}^{-1}\mathbf{U}^T\left(\frac{1}{p_i}\mathbf{a}_i\mathbf{a}_i^T - \mathbf{A}\mathbf{A}^T\right)\mathbf{U}\mathbf{\Sigma}^{-1} \text{ with probability } p_i,$$

have via (2.13) and (2.14), with probability 1:

$$-\frac{\epsilon^2}{32\log(\text{rank}(\mathbf{A})/\delta)}\mathbf{I} \preceq \mathbf{X}_j \prec \frac{\epsilon^2}{32\log(\text{rank}(\mathbf{A})/\delta)}\mathbf{I}.$$

Or equivalently, $\|\mathbf{X}_j\|_2 \leq \frac{\epsilon^2}{32\log(\text{rank}(\mathbf{A})/\delta)}$, which gives the claim. □

We next bound the variance of $\mathbf{Y}$.

Claim 2.2.10. $\mathbb{E}[\mathbf{Y}^2] \preceq \frac{\epsilon^2}{32\log(\text{rank}(\mathbf{A})/\delta)}\mathbf{I}$.

Proof. The proof is a simple but tedious calculation:

$$\begin{aligned}\mathbb{E}[\mathbf{Y}^2] = t\cdot\mathbb{E}[\mathbf{X}_j^2] = \frac{1}{t}\sum_i p_i \cdot \Bigg(&\frac{1}{p_i^2}\mathbf{\Sigma}^{-1}\mathbf{U}^T\mathbf{a}_i\mathbf{a}_i^T\mathbf{U}\mathbf{\Sigma}^{-2}\mathbf{U}^T\mathbf{a}_i\mathbf{a}_i^T\mathbf{U}\mathbf{\Sigma}^{-1}\\ &-2\frac{1}{p_i}\mathbf{\Sigma}^{-1}\mathbf{U}^T\mathbf{a}_i\mathbf{a}_i^T\mathbf{U}\mathbf{\Sigma}^{-2}\mathbf{U}^T\mathbf{A}\mathbf{A}^T\mathbf{U}\mathbf{\Sigma}^{-1} + \mathbf{\Sigma}^{-1}\mathbf{U}^T\mathbf{A}\mathbf{A}^T\mathbf{U}\mathbf{\Sigma}^{-2}\mathbf{U}^T\mathbf{A}\mathbf{A}^T\mathbf{U}\mathbf{\Sigma}^{-1}\Bigg)\\ &\preceq \frac{1}{t}\sum_i\left[\frac{\sum_i\tilde{\ell}_i}{\tilde{\ell}_i}\cdot\ell_i(\mathbf{A})\cdot\mathbf{\Sigma}^{-1}\mathbf{U}\mathbf{a}_i\mathbf{a}_i^T\mathbf{U}\mathbf{\Sigma}^{-1}\right] - \frac{1}{t}\mathbf{\Sigma}^{-1}\mathbf{U}^T\mathbf{A}\mathbf{A}^T\mathbf{U}\mathbf{\Sigma}^{-1}\\ &\preceq \frac{\epsilon^2}{32\log(\text{rank}(\mathbf{A})/\delta)}\mathbf{\Sigma}^{-1}\mathbf{U}^T\mathbf{A}\mathbf{A}^T\mathbf{U}\mathbf{\Sigma}^{-1} \preceq \frac{\epsilon^2}{32\log(\text{rank}(\mathbf{A})/\delta)}\mathbf{I}.\end{aligned}$$

□

We can plug the bounds of Claims 2.2.9 and 2.2.10 into Theorem 7.3.1 of [Tro15], to give for $\epsilon, \delta \in (0, 1/2]$:

$$\begin{aligned}
\mathbb{P}\left[\|\mathbf{Y}\|_2 \geq \epsilon/2\right] &\leq 4 \cdot \operatorname{rank}(\mathbf{A}) \cdot e^{\frac{-\epsilon^2/8}{\left(\frac{\epsilon^2}{32\log(\operatorname{rank}(\mathbf{A})/\delta)}(1+\epsilon/6)\right)}} \\
&\leq 4 \cdot \operatorname{rank}(\mathbf{A}) \cdot e^{-\frac{48}{13}\log(\operatorname{rank}(\mathbf{A})/\delta)} \\
&\leq \delta.
\end{aligned}$$

The last bound follows from the fact that

$$\begin{aligned}
\log(4\operatorname{rank}(\mathbf{A})/\delta) &= \log(4) + \log(\operatorname{rank}(\mathbf{A})/\delta) \\
&\leq 3\log(\operatorname{rank}(\mathbf{A})/\delta) \\
&< \frac{48}{13}\log(\operatorname{rank}(\mathbf{A})/\delta)
\end{aligned}$$

since $\operatorname{rank}(\mathbf{A})/\delta \geq 2$ by the restriction that $\delta \leq 1/2$. This gives the Lemma by Claim 2.2.7. □

Lemma 2.2.6 also gives an easy Corollary for the approximation obtained when sampling with ridge leverage scores. In comparison to Lemma 2.2.6 this approximation is weaker – since it included an additive $\epsilon\lambda\mathbf{I}$ term. However, as discussed, since the sum of ridge leverage scores is smaller than the sum of standard leverage scores, this approximation requires sampling fewer columns. Bounds similar to Corollary 2.2.11 have also appeared, for example, in [AM15b, CMM17, AKM+17].

Corollary 2.2.11 (Ridge Leverage Scores Sampling Spectral Approximation)**.** *For any $\mathbf{A} \in \mathbb{R}^{n\times d}$, for all $i \in [d]$, let $\tilde{\tau}_i^\lambda \geq \tau_i^\lambda(\mathbf{A})$ be an overestimate for $\mathbf{A}$'s i^{th} λ column ridge leverage score (Definition 2.2.1). Let $p_i = \frac{\tilde{\tau}_i^\lambda}{\sum_i \tilde{\tau}_i^\lambda}$ and for any $\epsilon, \delta \in (0, 1/2]$, $t \geq \frac{32\log(\operatorname{rank}(\mathbf{A})/\delta)}{\epsilon^2}\sum_i \tilde{\tau}_i^\lambda$. Construct $\mathbf{C}$ by sampling t columns of $\mathbf{A}$, each set to $\frac{1}{\sqrt{tp_i}}\mathbf{a}_i$ with probability p_i. With probability $\geq 1-\delta$:*

$$(1-\epsilon)\mathbf{C}\mathbf{C}^T - \epsilon\lambda\mathbf{I} \preceq \mathbf{A}\mathbf{A}^T \preceq (1+\epsilon)\mathbf{C}\mathbf{C}^T + \epsilon\lambda\mathbf{I}. \tag{2.15}$$

Proof. We can instantiate Lemma 2.2.6 with $[\mathbf{A}, \sqrt{\lambda}\mathbf{I}]$ setting $\tilde{\ell}_i = \tilde{\tau}_i^\lambda$. We simply fix the columns of the identity to appear in our sample. All calculations go through, and with probability $\geq 1-\delta$:

$$(1-\epsilon)[\mathbf{C}, \sqrt{\lambda}\mathbf{I}][\mathbf{C}, \sqrt{\lambda}\mathbf{I}]^T \preceq [\mathbf{A}, \sqrt{\lambda}\mathbf{I}][\mathbf{A}, \sqrt{\lambda}\mathbf{I}]^T \preceq (1+\epsilon)[\mathbf{C}, \sqrt{\lambda}\mathbf{I}][\mathbf{C}, \sqrt{\lambda}\mathbf{I}]^T,$$

which gives the desired bound if we subtract $\lambda\mathbf{I}$ from all sides. □

We note that it is possible to tighten Corollary 2.2.11. The rank($\mathbf{A}$) term in the log can be replaced by $\sum_i \tau_i^\lambda$, which, for example, for the rank-k ridge leverage scores of Definition 2.2.2 is bounded by $2k$ by Lemma 2.2.3. However, in the applications of this chapter, this stronger bound will not be necessary. See Theorem 5 of [CMM17] for a proof of the stronger bound.

Approximation Bounds for Regression and Low-Rank Approximation

Using the basic spectral approximation bounds of Lemma 2.2.6 and Corollary 2.2.11, it is possible to show that sampling by a matrix's (ridge) leverage scores gives a sketch of this matrix that can be used to approximately solve a number of fundamental low-rank approximation and regression problems.

For low-rank approximation, using the slight strengthening of Corollary 2.2.11 that we prove in [CMM17], it is possible to show that sampling by the rank-k ridge leverage scores yields a projection-cost-preserving sketch (PCP) of $\mathbf{A}$ (Definition 2.1.3):

Lemma 2.2.12 (Theorem 6 of [CMM17]). *For any $\mathbf{A} \in \mathbb{R}^{n\times d}$, for all $i \in [d]$, let $\tilde{\tau}_i^k \geq \tau_i^k(\mathbf{A})$ be an overestimate for $\mathbf{A}$'s i^{th} rank-k ridge leverage score (Definition 2.2.2). Let $p_i = \frac{\tilde{\tau}_i^k}{\sum_i \tilde{\tau}_i^k}$ and for any $\epsilon, \delta \in (0, 1/2]$, $t \geq \frac{c\log(k/\delta)}{\epsilon^2}\sum_i \tilde{\tau}_i^k$ for sufficiently large constant c. Construct $\mathbf{C}$ by sampling t columns of $\mathbf{A}$, each independently set to $\frac{1}{\sqrt{tp_i}}\mathbf{a}_i$ with probability p_i. Then with probability $\geq 1-\delta$, $\mathbf{C}$ as an (ϵ, k)-PCP of $\mathbf{A}$ (Definition 2.1.3).*

We do not prove Lemma 2.2.12 here, although its proof follows a similar pattern to the proofs of Lemmas 2.4.1 and 2.4.3, which we present in Section 2.4.

Since the distance $\|\mathbf{A}-\mathbf{PA}\|_F^2$ from $\mathbf{A}$ to any rank-k projection of $\mathbf{A}$ is preserved by $\mathbf{C}$, any near-optimal low-rank approximation of $\mathbf{C}$ yields a near-optimal low-rank approximation of $\mathbf{A}$. Further, $\mathbf{C}$ is much smaller than $\mathbf{A}$, so such a low-rank approximation can be computed quickly. Specifically, if each $\tilde{\tau}_i^k$ is an $O(1)$ approximation to $\tau_i^k(\mathbf{A})$, then by Lemma 2.2.3, $\sum_i \tau_i^k(\mathbf{A}) \leq 2k$ so $\sum_i \tilde{\tau}_i = O(k)$ and $\mathbf{C}$ has just $t = \tilde{O}(k/\epsilon^2)$ columns.

In addition to Lemma 2.2.12 we make use of a well known result on approximate regression via leverage score sampling, which we state and reprove here for completeness. See Theorem 38 of [CW13] and [DMM08] for earlier work showing similar results. In this result and throughout the rest of the chapter we write a subset of rows or columns of a matrix as the product of that matrix and a *weighted sampling matrix* defined below:

Definition 2.2.13 (Sampling Matrix)**.** *A weighted sampling matrix* $\mathbf{S} \in \mathbb{R}^{n \times d}$ *is any matrix with each row a scalar multiple of one of the standard basis vectors* $\mathbf{e}_1, ..., \mathbf{e}_n$*.*

We have:

Lemma 2.2.14 (Approximate Regression Via Leverage Score Sampling)**.** *For any* $\mathbf{A} \in \mathbb{R}^{n \times d}$ *and* $\mathbf{Y} \in \mathbb{R}^{n \times d'}$*, for all* $i \in [n]$ *let* $\tilde{\ell}_i \geq \ell_i(\mathbf{A})$ *be an overestimate for* $\mathbf{A}$*'s* i^{th} *row leverage score (Definition 2.2.4). Let* $p_i = \frac{\tilde{\ell}_i}{\sum_i \tilde{\ell}_i}$ *and for* $\epsilon, \delta \in (0, 1/2]$*, let* $t = c\left(\log(\text{rank}(\mathbf{A})) + \frac{1}{\delta \cdot \epsilon}\right) \cdot \sum_i \tilde{\ell}_i$ *for some sufficiently large constant* c*. Let* $\mathbf{S} \in \mathbb{R}^{n \times t}$ *be the sampling matrix whose* i^{th} *column is set independently to* $\frac{1}{\sqrt{tp_i}}\mathbf{e}_i$ *with probability* p_i*. Then with probability* $\geq 1 - \delta$*, letting*

$$\tilde{\mathbf{X}} = \operatorname*{arg\,min}_{\mathbf{X} \in \mathbb{R}^{d \times d'}} \|\mathbf{S}^T\mathbf{A}\mathbf{X} - \mathbf{S}^T\mathbf{Y}\|_F^2,$$

we have:

$$\|\mathbf{A}\tilde{\mathbf{X}} - \mathbf{Y}\|_F^2 \leq (1 + \epsilon) \min_{\mathbf{X} \in \mathbb{R}^{d \times d'}} \|\mathbf{A}\mathbf{X} - \mathbf{Y}\|_F^2.$$

Proof. We first note that we can assume without loss of generality that $\mathbf{A}$ has orthonormal columns. This is because if the lemma holds for $\mathbf{A}$ with orthonormal columns, for general $\mathbf{A}$, we can consider an orthonormal span $\mathbf{U} \in \mathbb{R}^{n \times \text{rank}(\mathbf{A})}$ for the columns of $\mathbf{A}$. We can write $\mathbf{A}\tilde{\mathbf{X}} = \mathbf{U}\tilde{\mathbf{Z}}$ for some $\tilde{\mathbf{Z}} \in \mathbb{R}^{\text{rank}(\mathbf{A}) \times d'}$ and have

$$\tilde{\mathbf{Z}} = \operatorname*{arg\,min}_{\mathbf{X} \in \mathbb{R}^{\text{rank}(\mathbf{A}) \times d'}} \|\mathbf{S}^T\mathbf{U}\mathbf{X} - \mathbf{S}^T\mathbf{Y}\|_F^2.$$

Further, by Lemma 2.2.5, the leverage scores of $\mathbf{U}$ are equal to those of $\mathbf{A}$ and so, with probability $\geq 1 - \delta$, we have:

$$\|\mathbf{U}\tilde{\mathbf{Z}} - \mathbf{Y}\|_F^2 \leq (1 + \epsilon) \min_{\mathbf{X} \in \mathbb{R}^{\text{rank}(\mathbf{A}) \times d'}} \|\mathbf{U}\mathbf{X} - \mathbf{Y}\|_F^2. \tag{2.16}$$

Again since $\mathbf{U}$ spans the columns of $\mathbf{A}$ we have:

$$\min_{\mathbf{X} \in \mathbb{R}^{\text{rank}(\mathbf{A}) \times d'}} \|\mathbf{U}\mathbf{X} - \mathbf{Y}\|_F^2 = \min_{\mathbf{X} \in \mathbb{R}^{d \times d'}} \|\mathbf{A}\mathbf{X} - \mathbf{Y}\|_F^2.$$

So (2.16) combined with the fact that $\mathbf{U}\tilde{\mathbf{Z}} = \mathbf{A}\tilde{\mathbf{X}}$ gives:

$$\|\mathbf{A}\tilde{\mathbf{X}} - \mathbf{Y}\|_F^2 \leq (1 + \epsilon) \min_{\mathbf{X} \in \mathbb{R}^{d \times d'}} \|\mathbf{A}\mathbf{X} - \mathbf{Y}\|_F^2,$$

which gives the lemma.

Proof for Orthonormal A. Thus, we prove the lemma for $\mathbf{A}$ with orthonormal columns. Let $\mathbf{X}^* = \arg\min_{\mathbf{X}\in\mathbb{R}^{d\times d'}} \|\mathbf{AX}-\mathbf{Y}\|_F^2$. We first have:

Claim 2.2.15. *With probability* $\geq 1-\delta$, *both:*

1. $\frac{1}{2}\mathbf{A}^T\mathbf{SS}^T\mathbf{A} \preceq \mathbf{A}^T\mathbf{A} \preceq \frac{3}{2}\mathbf{A}^T\mathbf{SS}^T\mathbf{A}$.

2. $\|\mathbf{A}^T\mathbf{SS}^T(\mathbf{Y}-\mathbf{AX}^*)\|_F \leq \sqrt{\epsilon/2}\|\mathbf{Y}-\mathbf{AX}^*\|_F$.

Proof. Since $\tilde{\ell}_i \geq \ell_i(\mathbf{A})$ and since $t \geq c\log(\text{rank}(\mathbf{A})/\delta)\cdot\sum_i \tilde{\ell}_i$, by Lemma 2.2.6, if c is set large enough, the first bound holds with probability $\geq 1-\delta/2$. To prove the second bound, we apply a standard approximate matrix multiplication result. We reprove this result here for completeness.

Lemma 2.2.16 (Approximate Matrix Multiplication, based on [DKM06])**.** *Given* $\mathbf{M}\in\mathbb{R}^{n\times d}$ *and* $\mathbf{N}\in\mathbb{R}^{p\times d}$ *with columns* $\mathbf{m}_i$ *and* $\mathbf{n}_i$ *respectively, for* $i\in[d]$ *let* $\tilde{\gamma}_i \geq \|\mathbf{m}_i\|_2^2$ *be an overestimate for the* i^{th} *squared column norm of* $\mathbf{M}$. *Let* $p_i = \frac{\tilde{\gamma}_i}{\sum_i \tilde{\gamma}_i}$, *and* t *satisfy* $t \geq \frac{1}{\delta\cdot\epsilon^2}\cdot\sum_i\tilde{\gamma}_i$. *Construct weighted sampling matrix* $\mathbf{S}\in\mathbb{R}^{d\times t}$, *whose* j^{th} *column is set to* $\frac{1}{\sqrt{tp_i}}\mathbf{e}_i$ *with probability* p_i. *Then, with probability* $\geq 1-\delta$,

$$\|\mathbf{MN}^T - \mathbf{MSS}^T\mathbf{N}^T\|_F \leq \epsilon\|\mathbf{N}\|_F.$$

Proof. We will bound $\mathbb{E}\left[\|\mathbf{MN}^T-\mathbf{MSS}^T\mathbf{N}^T\|_F^2\right]$ and then apply Markov's inequality to give the theorem. Applying Lemma 4 of [DKM06] we have:

$$\begin{aligned}\mathbb{E}\left[\|\mathbf{MN}^T-\mathbf{MSS}^T\mathbf{N}^T\|_F^2\right] &= \sum_{i=1}^n \frac{\|\mathbf{m}_i\|_2^2\|\mathbf{n}_i\|_2^2}{tp_i} - \frac{1}{t}\|\mathbf{MN}^T\|_F^2\\ &\leq \sum_{i=1}^n \frac{\|\mathbf{m}_i\|_2^2\|\mathbf{n}_i\|_2^2}{\frac{1}{\delta\epsilon^2}\cdot\tilde{\gamma}_i},\end{aligned}$$

which by the assumption that for all $i\in[d]$, $\tilde{\gamma}_i \geq \|\mathbf{m}_i\|_2^2$ gives:

$$\mathbb{E}\left[\|\mathbf{MN}^T-\mathbf{MSS}^T\mathbf{N}^T\|_F^2\right] \leq \delta\epsilon^2\|\mathbf{N}\|_F^2.$$

Thus, by Markov's inequality, with probability $\geq 1-\delta$,

$$\|\mathbf{MN}^T-\mathbf{MSS}^T\mathbf{N}^T\|_F^2 \leq \epsilon^2\|\mathbf{N}\|_F^2$$

and so $\|\mathbf{MN}^T-\mathbf{MSS}^T\mathbf{N}^T\|_F \leq \epsilon\|\mathbf{N}\|_F$, giving the lemma. □

Since $\mathbf{A}$ is orthonormal, by Lemma 2.2.5 we have $\ell_i(\mathbf{A}) = \|\mathbf{a}_i\|_2^2$, where $\mathbf{a}_i$ is the i^{th} row of $\mathbf{A}$. Thus by Lemma 2.2.16, since $t \geq \frac{c}{\delta \cdot \epsilon} \cdot \sum_i \tilde{\ell}_i$, if c is large enough, with probability $\geq 1 - \delta/2$,

$$\|\mathbf{A}^T\mathbf{S}\mathbf{S}^T(\mathbf{A}\mathbf{X}^* - \mathbf{Y}) - \mathbf{A}^T(\mathbf{A}\mathbf{X}^* - \mathbf{Y})\|_F \leq \sqrt{\epsilon/2}\|\mathbf{Y} - \mathbf{A}\mathbf{X}^*\|_F.$$

This gives the second bound of Claim 2.2.15 after noting that

$$\mathbf{A}^T(\mathbf{A}\mathbf{X}^* - \mathbf{Y}) = \mathbf{A}^T(\mathbf{A}\mathbf{A}^T\mathbf{Y} - \mathbf{Y}) = \mathbf{A}^T\mathbf{Y} - \mathbf{A}^T\mathbf{Y} = \mathbf{0}.$$

By a union bound, both the first and second bounds hold with probability $\geq 1 - \delta$, completing the claim. □

Now, since $\mathbf{A}$ has orthonormal columns, $\mathbf{X}^* = \mathbf{A}^T\mathbf{Y}$. We can write $\mathbf{A}\tilde{\mathbf{X}} - \mathbf{Y} = (\mathbf{A}\mathbf{X}^* - \mathbf{Y}) + \mathbf{A}(\tilde{\mathbf{X}} - \mathbf{X}^*)$. $(\mathbf{A}\mathbf{X}^* - \mathbf{Y}) = (\mathbf{A}\mathbf{A}^T - \mathbf{I})\mathbf{A}$ and so has columns orthogonal to the column span of $\mathbf{A}$. So by Pythagorean theorem:

$$\|\mathbf{A}\tilde{\mathbf{X}} - \mathbf{Y}\|_F^2 = \|\mathbf{A}\mathbf{X}^* - \mathbf{Y}\|_F^2 + \|\mathbf{A}(\tilde{\mathbf{X}} - \mathbf{X}^*)\|_F^2.$$

Thus to show the lemma, it suffices to prove that

$$\|\mathbf{A}(\tilde{\mathbf{X}} - \mathbf{X}^*)\|_F^2 \leq \epsilon\|\mathbf{A}\mathbf{X}^* - \mathbf{Y}\|_F^2. \tag{2.17}$$

Since $\mathbf{A}$ has orthonormal columns we have $\|\mathbf{A}(\tilde{\mathbf{X}} - \mathbf{X}^*)\|_F^2 = \|\tilde{\mathbf{X}} - \mathbf{X}^*\|_F^2$. Further, by conclusion (1) of Claim 2.2.15, since $\mathbf{A}^T\mathbf{A} = \mathbf{I}$, we have $\mathbf{I} \preceq \frac{3}{2}\mathbf{A}^T\mathbf{S}\mathbf{S}^T\mathbf{A}$. This gives:

$$\|\mathbf{A}(\tilde{\mathbf{X}} - \mathbf{X}^*)\|_F^2 \leq \frac{3}{2} \cdot \|\mathbf{A}^T\mathbf{S}\mathbf{S}^T\mathbf{A}(\tilde{\mathbf{X}} - \mathbf{X}^*)\|_F^2. \tag{2.18}$$

Since $\tilde{\mathbf{X}} = \arg\min_{\mathbf{X} \in \mathbb{R}^{d \times d'}} \|\mathbf{S}^T\mathbf{A}\mathbf{X} - \mathbf{S}^T\mathbf{Y}\|_F^2$, the columns of $\mathbf{S}^T\mathbf{Y} - \mathbf{S}^T\mathbf{A}\tilde{\mathbf{X}}$ are orthogonal to the columns of $\mathbf{S}^T\mathbf{A}$. Thus we can write:

$$\begin{aligned}
\|\mathbf{A}^T\mathbf{S}\mathbf{S}^T\mathbf{A}(\tilde{\mathbf{X}} - \mathbf{X}^*)\|_F^2 &= \|\mathbf{A}^T\mathbf{S}(\mathbf{S^T}\mathbf{A}\tilde{\mathbf{X}} - \mathbf{S}^T\mathbf{Y} + \mathbf{S}^T\mathbf{Y} - \mathbf{S}^T\mathbf{A}\mathbf{X}^*)\|_F^2 \\
&= \|\mathbf{A}^T\mathbf{S}(\mathbf{S}^T\mathbf{Y} - \mathbf{S}^T\mathbf{A}\mathbf{X}^*)\|_F^2 \\
&= \|\mathbf{A}^T\mathbf{S}\mathbf{S}^T(\mathbf{Y} - \mathbf{A}\mathbf{X}^*)\|_F^2.
\end{aligned}$$

By conclusion (2) of 2.2.15 we have $\|\mathbf{A}^T\mathbf{S}\mathbf{S}^T(\mathbf{Y} - \mathbf{A}\mathbf{X}^*)\|_F^2 \leq \frac{\epsilon}{2}\|\mathbf{Y} - \mathbf{A}\mathbf{X}^*\|_F^2$. Combined with (2.18) this gives:

$$\|\mathbf{A}(\tilde{\mathbf{X}} - \mathbf{X}^*)\|_F^2 \leq \frac{3}{2} \cdot \frac{\epsilon}{2} \|\mathbf{Y} - \mathbf{A}\mathbf{X}^*\|_F^2 \leq \epsilon \|\mathbf{Y} - \mathbf{A}\mathbf{X}^*\|_F^2,$$

which gives (2.17) and thus completes the lemma. □

2.2.3 Fast Ridge Leverage Score Approximation

With Lemma 2.2.12 established, the difficulty remains in computing the approximate leverage scores. To do this, we use our main result from [MM17], which applies to computing the leverage scores of $\mathbf{A}^{1/2}$ for any PSD $\mathbf{A}$. Recall that letting the SVD of $\mathbf{A}$ be $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{U}^T$, we define $\mathbf{A}^{1/2} = \mathbf{U}\mathbf{\Sigma}^{1/2}\mathbf{U}^T$.

Lemma 2.2.17 (Corollary of Theorem 20 of [MM17])**.** *There is an algorithm that given any PSD matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$, $k \in \mathbb{Z}^{\geq 1}$, and $\delta \in (0, 1/2]$ runs in $O(n(k \log(k/\delta))^{\bar{\omega}-1})$ time, accesses $O(nk \log(k/\delta))$ entries of $\mathbf{A}$, and returns for each $i \in [n]$, $\tilde{\tau}_i^k(\mathbf{A}^{1/2})$ such that with probability $\geq 1 - \delta$, for all i:*

$$\tau_i^k(\mathbf{A}^{1/2}) \leq \tilde{\tau}_i^k(\mathbf{A}^{1/2}) \leq 3\tau_i^k(\mathbf{A}^{1/2}).$$

Proof. In Theorem 20 of [MM17] we show that by using a recursive ridge leverage score sampling algorithm, it is possible to return (with probability $\geq 1-\delta$) a sampling matrix $\mathbf{S} \in \mathbb{R}^{n \times s}$ with $s = O\left(k \log(k/\delta)\right)$ such that, letting $\lambda = \frac{1}{k}\|\mathbf{A}^{1/2} - \mathbf{A}_k^{1/2}\|_F^2$:

$$\frac{1}{2}\left(\mathbf{A} + \lambda \mathbf{I}\right) \preceq \left(\mathbf{A}^{1/2}\mathbf{S}\mathbf{S}^T\mathbf{A}^{1/2} + \lambda \mathbf{I}\right) \preceq \frac{3}{2}\left(\mathbf{A} + \lambda \mathbf{I}\right). \tag{2.19}$$

If we set $\tilde{\tau}_i^k(\mathbf{A}^{1/2}) = 2 \cdot \mathbf{x}_i^T \left(\mathbf{A}^{1/2}\mathbf{S}\mathbf{S}^T\mathbf{A}^{1/2} + \lambda \mathbf{I}\right)^+ \mathbf{x}_i$, where $\mathbf{x}_i$ is the i^{th} column of $\mathbf{A}^{1/2}$ we have the desired bound. Of course, we cannot directly compute this value without factoring $\mathbf{A}$ to form $\mathbf{A}^{1/2}$. However, as we show in Lemma 6 of [MM17], for $\lambda > 0$: [10]

$$\mathbf{x}_i^T \left(\mathbf{A}^{1/2}\mathbf{S}\mathbf{S}^T\mathbf{A}^{1/2} + \lambda \mathbf{I}\right)^{-1} \mathbf{x}_i = \frac{1}{\lambda}\left(\mathbf{A} - \mathbf{A}\mathbf{S}(\mathbf{S}^T\mathbf{A}\mathbf{S} + \lambda \mathbf{I})^{-1}\mathbf{S}^T\mathbf{A}\right)_{i,i}. \tag{2.20}$$

Computing $(\mathbf{S}^T\mathbf{A}\mathbf{S} + \lambda \mathbf{I})^{-1}$ requires $O(s^2) = O((k \log(k/\delta))^2)$ accesses to $\mathbf{A}$ and $O(s^{\bar{\omega}}) = O((k \log(k/\delta))^{\bar{\omega}})$ time. Computing all n diagonal entries of $\mathbf{A}\mathbf{S}(\mathbf{S}^T\mathbf{A}\mathbf{S} +$

[10] For the rank-k ridge leverage scores, $\lambda > 0$ unless $\mathbf{A}$ has rank $\leq k$. In this case $\lambda = 0$ and (2.20) does not hold. However, by (2.19), the columns of $\mathbf{S}$ already span the column space of $\mathbf{A}$. Thus, we can compute $\mathbf{M} \in \mathbb{R}^{n \times k}$ whose columns span those of $\mathbf{S}$ and then directly project to these columns to compute an optimal low-rank approximation of $\mathbf{A}$ as in Steps 7-8 of Algorithm 2.5.1.

$\lambda I)^{-1}\mathbf{S}^T\mathbf{A}$ then requires $O(nk\log(k/\delta))$ accesses to $\mathbf{A}$ and $O(n(k\log(k/\delta))^{\bar{\omega}-1})$ time. With these entries in hand we can simply subtract from the diagonal entries of $\mathbf{A}$ and rescale to give the final leverage score approximation. Critically, this calculation always reads *all diagonal entries* of $\mathbf{A}$, allowing it to identify rows containing large off-diagonal entries and skirt the $\text{nnz}(\mathbf{A})$ time lower bound for general matrices.

Note that the stated runtime in [MM17] for outputting $\mathbf{S}$ is $\tilde{O}(nk)$ accesses to $\mathbf{A}$ (*kernel evaluations* in the language of [MM17]) and $\tilde{O}(nk^2)$ runtime. However this runtime is improved to $\tilde{O}(nk^{\bar{\omega}-1})$ using fast matrix multiplication. □

2.3 Column Sampling

To apply Lemmas 2.2.12 and 2.2.17 to low-rank approximation of $\mathbf{A}$, we now show that the ridge leverage scores of $\mathbf{A}^{1/2}$ coarsely upper bound those of $\mathbf{A}$. As shown in Corollary 2.3.3, we can then sample columns of $\mathbf{A}$ by these scores to obtain an (ϵ, k)-PCP $\mathbf{C}$, which can be further sampled to efficiently obtain a near-optimal low-rank approximation of $\mathbf{A}$. We note that the idea of simultaneously sampling rows and columns from $\mathbf{A}$ to form an approximation is analogous to existing work on CUR matrix approximation [DMM08, MD09] and Nyström approximation of positive semidefinite matrices [WS01, GM13].

Lemma 2.3.1 (Ridge Leverage Score Bound)**.** *For any PSD matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$ and any $k \leq n$:*

$$\tau_i^k(\mathbf{A}) \leq 2\sqrt{\frac{n}{k}} \cdot \tau_i^k(\mathbf{A}^{1/2}).$$

Proof. We write $\mathbf{A}^{1/2}$ in its singular value decomposition $\mathbf{A}^{1/2} = \mathbf{U}\mathbf{\Sigma}^{1/2}\mathbf{U}^T$. Letting $\mathbf{x}_i$ denote the i^{th} column of $\mathbf{A}^{1/2}$ we have:

$$\begin{aligned}\tau_i^k(\mathbf{A}^{1/2}) &= \mathbf{x}_i^T\left(\mathbf{A} + \frac{\|\mathbf{A}^{1/2} - \mathbf{A}_k^{1/2}\|_F^2}{k}\mathbf{I}\right)^{-1}\mathbf{x}_i \\ &= \mathbf{x}_i^T\mathbf{U}\bar{\mathbf{\Sigma}}\mathbf{U}^T\mathbf{x}_i, \qquad (2.21)\end{aligned}$$

where $\bar{\mathbf{\Sigma}}_{i,i} \overset{\text{def}}{=} \frac{1}{\sigma_i(\mathbf{A})+\frac{1}{k}\sum_{j=k+1}^{n}\sigma_j(\mathbf{A})}$. We can similarly write:

$$\begin{aligned}\tau_i^k(\mathbf{A}) &= \mathbf{a}_i^T\left(\mathbf{A}^2+\frac{\|\mathbf{A}-\mathbf{A}_k\|_F^2}{k}\mathbf{I}\right)^{-1}\mathbf{a}_i\\ &= \mathbf{x}_i^T\mathbf{A}^{1/2}\left(\mathbf{A}^2+\frac{\|\mathbf{A}-\mathbf{A}_k\|_F^2}{k}I\right)^{-1}\mathbf{A}^{1/2}\mathbf{x}_i\\ &= \mathbf{x}_i^T\mathbf{U}\hat{\mathbf{\Sigma}}\mathbf{U}^T\mathbf{x}_i, \end{aligned} \tag{2.22}$$

where $\hat{\mathbf{\Sigma}}_{i,i} \overset{\text{def}}{=} \frac{\sigma_i(\mathbf{A})}{\sigma_i(\mathbf{A})^2+\frac{1}{k}\sum_{j=k+1}^{n}\sigma_j(\mathbf{A})^2}$. We have:

Claim 2.3.2. *If $\hat{\mathbf{\Sigma}} \preceq 2\sqrt{\frac{n}{k}}\cdot\bar{\mathbf{\Sigma}}$ then for all i, $\tau_i^k(\mathbf{A}) \le 2\sqrt{\frac{n}{k}}\cdot\tau_i^k(\mathbf{A}^{1/2})$.*

Proof. $\hat{\mathbf{\Sigma}} \preceq 2\sqrt{\frac{n}{k}}\cdot\bar{\mathbf{\Sigma}}$ gives that for any $\mathbf{x}\in\mathbb{R}^{\text{rank}(\mathbf{A})}$,

$$\mathbf{x}^T\hat{\mathbf{\Sigma}}\mathbf{x} \le 2\sqrt{\frac{n}{k}}\cdot\mathbf{x}^T\bar{\mathbf{\Sigma}}\mathbf{x}.$$

For any i, setting $\mathbf{x}=\mathbf{U}^T\mathbf{x}_i$ and applying (2.21) and (2.22) then gives

$$\tau_i^k(\mathbf{A}) \le 2\sqrt{\frac{n}{k}}\cdot\tau_i^k(\mathbf{A}^{1/2}).$$

□

By Claim 2.3.2 it just remains to show $\hat{\mathbf{\Sigma}} \preceq 2\sqrt{\frac{n}{k}}\cdot\bar{\mathbf{\Sigma}}$ to give the lemma. Specifically we must show, for all i, $\hat{\mathbf{\Sigma}}_{i,i} \le 2\sqrt{\frac{n}{k}}\cdot\bar{\mathbf{\Sigma}}_{i,i}$. After cross-multiplying, this is equivalent to:

$$\sigma_i(\mathbf{A})^2+\frac{1}{k}\cdot\sigma_i(\mathbf{A})\cdot\sum_{j=k+1}^{n}\sigma_j(\mathbf{A}) \le 2\sqrt{\frac{n}{k}}\left(\sigma_i(\mathbf{A})^2+\frac{1}{k}\sum_{j=k+1}^{n}\sigma_j(\mathbf{A})^2\right). \tag{2.23}$$

We show (2.23) for every i by considering two cases:

Case 1: First consider the relatively large singular values. Say we have $\frac{1}{k}\sum_{j=k+1}^{n}\sigma_j(\mathbf{A}) \le \sqrt{\frac{n}{k}}\cdot\sigma_i(\mathbf{A})$. Then:

$$\sigma_i(\mathbf{A})^2+\frac{1}{k}\cdot\sigma_i(\mathbf{A})\cdot\sum_{j=k+1}^{n}\sigma_j(\mathbf{A}) \le \left(1+\sqrt{\frac{n}{k}}\right)\sigma_i(\mathbf{A})^2,$$

which gives (2.23) since $(1+\sqrt{n/k}) \ge 2$.

Case 2: Next consider small singular values with $\frac{1}{k}\sum_{j=k+1}^{n}\sigma_j(\mathbf{A}) \geq \sqrt{\frac{n}{k}} \cdot \sigma_i(\mathbf{A})$. In this case:

$$\begin{aligned}
\sigma_i(\mathbf{A})^2 + \frac{1}{k}\cdot\sigma_i(\mathbf{A})\cdot\sum_{j=k+1}^{n}\sigma_j(\mathbf{A}) &\leq \sigma_i(\mathbf{A})^2 + \frac{1}{\sqrt{n}\cdot k^{3/2}}\left(\sum_{j=k+1}^{n}\sigma_j(\mathbf{A})\right)^2 \\
&\leq \sigma_i(\mathbf{A})^2 + \frac{1}{\sqrt{n}\cdot k^{3/2}}\cdot n\sum_{j=k+1}^{n}\sigma_j(\mathbf{A})^2 \\
&\qquad\qquad\qquad\qquad \text{(Norm bound: } \|\cdot\|_1^2 \leq n\|\cdot\|_2^2) \\
&\leq \sqrt{\frac{n}{k}}\left(\sigma_i(\mathbf{A})^2 + \frac{1}{k}\sum_{j=k+1}^{n}\sigma_j(\mathbf{A})^2\right),
\end{aligned}$$

which gives (2.23), completing the proof. □

Combining Lemmas 2.2.3, 2.2.12, 2.2.17, 2.3.1 we have our first column sampling result for PSD matrices.

Corollary 2.3.3 (Fast PSD Ridge Leverage Score Sampling)**.** *There is an algorithm that given any PSD matrix* $\mathbf{A} \in \mathbb{R}^{n\times n}$*,* $k \in \mathbb{Z}^{\geq 1}$*, and* $\epsilon, \delta \in (0, 1/2]$*, runs in* $\tilde{O}(nk^{\bar{\omega}-1})$ *time, accesses* $\tilde{O}(nk)$ *entries of* $\mathbf{A}$*, and with probability* $\geq 1-\delta$ *outputs a weighted sampling matrix* $\mathbf{S}_1 \in \mathbb{R}^{n\times\tilde{O}\left(\frac{\sqrt{nk}}{\epsilon^2}\right)}$ *such that* $\mathbf{C} = \mathbf{A}\mathbf{S}_1$ *is an* (ϵ,k)*-PCP of* $\mathbf{A}$*.*

Proof. By Lemma 2.2.17 we can compute constant factor approximations to the ridge leverage scores of $\mathbf{A}^{1/2}$ in time $\tilde{O}(nk^{\bar{\omega}-1})$. Applying Lemma 2.3.1, if we scale these scores up by $2\sqrt{n/k}$ they will be overestimates of the ridge leverage scores of $\mathbf{A}$. If we set $t = c \cdot \frac{\log(k/\delta)}{\epsilon^2} \cdot \sum \tilde{\tau}_i^k$ for sufficiently large c, and generate $\mathbf{S}_1 \in \mathbb{R}^{n\times t}$ by sampling t columns of $\mathbf{A}$ with probabilities proportional to these estimated scores, by Lemma 2.2.12, $\mathbf{A}\mathbf{S}_1$ will be an (ϵ, k)-PCP of $\mathbf{A}$ with probability $\geq 1-\delta$. By Lemma 2.2.3, $\sum_{i=1}^{n}\tau_i^k(\mathbf{A}^{1/2}) \leq 2k$. So we have $t = \tilde{O}(\sum\tilde{\tau}_i^k/\epsilon^2) = \tilde{O}(\sqrt{nk}/\epsilon^2)$. □

Forming $\mathbf{A}\mathbf{S}_1$ requires reading just $\tilde{O}(n^{3/2}\sqrt{k}/\epsilon^2)$ entries of $\mathbf{A}$. At this point, we could employ any input sparsity time algorithm to find a near-optimal rank-k projection $\mathbf{P}$ for approximating $\mathbf{A}\mathbf{S}_1$ in $O(\text{nnz}(\mathbf{A}\mathbf{S}_1)) + n\,\text{poly}(k/\epsilon) = n^{3/2}\cdot\text{poly}(k/\epsilon)$ time. This would in turn yield a near-optimal low-rank approximation of $\mathbf{A}$. However, as we will see in the next section, by further sampling the rows of $\mathbf{A}\mathbf{S}_1$, we can significantly improve the n dependence in this runtime.

2.4 Row Sampling

To achieve near linear dependence on n, our algorithm will sample roughly $\sqrt{nk}$ rows from $\mathbf{AS}_1$, producing an even smaller matrix $\mathbf{S}_2^T\mathbf{AS}_1$ that we can afford to fully read and from which we can form a near-optimal low-rank approximation to $\mathbf{AS}_1$ and consequently to $\mathbf{A}$ (since $\mathbf{AS_1}$ will be an (ϵ, k)-PCP of $\mathbf{A}$ with high probability). However, sampling rows from $\mathbf{AS}_1$ is challenging: we cannot employ input sparsity time methods as we cannot afford to read the full matrix – it has dimensions $n \times \tilde{\Theta}\left(\frac{\sqrt{nk}}{\epsilon^2}\right)$ and so contains up to $\Omega(n^{3/2})$ nonzero entries. Further, since $\mathbf{AS}_1$ is no longer PSD, we cannot apply the same approach we used for $\mathbf{A}$, approximating the ridge leverage scores with those of $\mathbf{A}^{1/2}$ (which can be computed very efficiently via Lemma 2.2.17).

2.4.1 Approximating the Ridge Leverage Scores of $\mathbf{AS}_1$

Rewriting the formula for the rank-k ridge leverage scores of $\mathbf{AS}_1$ (Definition 2.2.2) using the SVD $\mathbf{AS}_1 = \mathbf{U\Sigma V}^T$ (and transposing $\mathbf{AS}_1$ to give row instead of column scores) we see that the row rank-k ridge leverage scores are the diagonal entries of:

$$\mathbf{AS}_1\left(\mathbf{S}_1^T\mathbf{A}^T\mathbf{AS}_1 + \frac{\|\mathbf{AS}_1 - (\mathbf{AS}_1)_k\|_F^2}{k}\mathbf{I}\right)^+ \mathbf{S}_1^T\mathbf{A}^T = \mathbf{U}\bar{\mathbf{\Sigma}}\mathbf{U}^T,$$

where $\bar{\mathbf{\Sigma}}$ is the diagonal matrix with:

$$\bar{\mathbf{\Sigma}}_{i,i} = \frac{\sigma_i(\mathbf{AS}_1)^2}{\sigma_i(\mathbf{AS}_1)^2 + \frac{\|\mathbf{AS}_1 - (\mathbf{AS}_1)_k\|_F^2}{k}}.$$

We can see from this computation that the row ridge leverage scores depend only on the column span $\mathbf{U}$ of $\mathbf{AS}_1$ and its singular value spectrum. Since $\mathbf{AS}_1$ is a column PCP of $\mathbf{A}$ this gives hope that the two matrices have similar row ridge leverage scores, both of which can be approximated using the leverage scores of $\mathbf{A}^{1/2}$.

Unfortunately, this is *not the case*. It is possible to have rows in $\mathbf{AS}_1$ with ridge leverage scores significantly higher than in $\mathbf{A}$. Thus, even if we knew the ridge leverage scores of $\mathbf{A}$, we would have to scale them up significantly to sample from $\mathbf{AS}_1$. As an example, consider $\mathbf{A}$ with relatively uniform rank-k ridge leverage scores: $\tau_i^k(\mathbf{A}) \approx k/n$ for all i. When a column is selected to be included in $\mathbf{AS}_1$ in the sampling scheme of Lemma 2.2.12 it will be reweighted by roughly a factor of $\sqrt{n/k}$. Now, append a large number of rows to $\mathbf{A}$ each with very small norm and just a containing single non-zero entry. These rows will have little effect on the ridge leverage scores if their norms are small enough. However, if the column corresponding to the nonzero in a row is selected, the row will appear in $\mathbf{AS}_1$ with $\sqrt{n/k}$ times the weight that it appears in $\mathbf{A}$, and its ridge leverage score will be roughly a factor n/k times higher.

2.4.2 Projection-Cost-Preserving Row Sampling

Fortunately, while we cannot approximate the row ridge leverage scores of $\mathbf{AS}_1$, we are still able to show that sampling the rows of $\mathbf{AS}_1$ by the rank-k' leverage scores of $\mathbf{A}^{1/2}$ scaled up by a $\sqrt{n/k'}$ factor for $k' = O(k/\epsilon)$ yields a row PCP for this matrix (of the form introduced in Definition 2.1.4). Our proof works not with the ridge scores of $\mathbf{AS}_1$ but with the *standard leverage scores* of a near-optimal low-rank approximation to this matrix – specifically the approximation given by projecting onto the top singular vectors of $\mathbf{A}$.

Note that our bound *does not show* that the rank-k' leverage scores of $\mathbf{A}^{1/2}$ actually approximate any the rank-k ridge leverage scores of $\mathbf{AS}_1$. However, we can still show the needed PCP property, which gives error in the Frobenius norm distance from any rank-k projection, of the form in Definition 2.1.4.

Lemma 2.4.1 (Frobenius Row PCP)**.** *For any PSD $\mathbf{A} \in \mathbb{R}^{n\times n}$ and $\epsilon, \delta \in (0, 1/2]$ let $k' = \lceil ck/\epsilon \rceil$ and for all $i \in [n]$ let $\tilde{\tau}_i^{k'}(\mathbf{A}^{1/2}) \geq \tau_i^{k'}(\mathbf{A}^{1/2})$ be an overestimate for the i^{th} rank-k' ridge leverage score of $\mathbf{A}^{1/2}$. Let $\tilde{\ell}_i = \sqrt{\frac{16n}{k'}} \cdot \tilde{\tau}_i^{k'}(\mathbf{A}^{1/2})$, $p_i = \frac{\tilde{\ell}_i}{\sum_i \tilde{\ell}_i}$, and $t \geq \frac{c' \log n}{\delta\cdot\epsilon^2}\sum_i \tilde{\ell}_i$. Construct weighted sampling matrices $\mathbf{S}_1, \mathbf{S}_2 \in \mathbb{R}^{n\times t}$ each whose j^{th} column is set to $\frac{1}{\sqrt{tp_i}}\mathbf{e}_i$ with probability p_i.*

For sufficiently large constants c, c', with probability $\geq 1-\delta$, we have:

- *$\mathbf{AS}_1$ is an (ϵ, k)-PCP of $\mathbf{A}$ (Definition 2.1.3).*

- *$\tilde{\mathbf{A}} = \mathbf{S}_2^T\mathbf{AS}_1$ is an $(\epsilon, 18/\delta, k)$-PCP (Definition 2.1.4) of $\mathbf{AS}_1$.*

By Lemma 2.2.3, the sum of rank-k' ridge leverage scores can be bounded by

$$\sum_i \tau_i^{k'}(\mathbf{A}^{1/2}) = O(k/\epsilon). \tag{2.24}$$

So if each $\tilde{\tau}_i^{k'}(\mathbf{A}^{1/2})$ is a constant factor approximation to $\tau_i^{k'}(\mathbf{A}^{1/2})$, we can set:

$$\begin{aligned} t &= \frac{c' \log n}{\delta \cdot \epsilon^2} \cdot \sum_i \tilde{\ell}_i \\ &= \frac{c' \log n}{\delta \cdot \epsilon^2} \cdot \sum_i \left(\sqrt{\frac{16n\epsilon}{k}} \tilde{\tau}_i^{k'}(\mathbf{A}^{1/2}) \right) \\ &= O\left(\frac{\log n}{\delta \cdot \epsilon^{1.5}} \cdot \sqrt{\frac{n}{k}} \cdot \sum_i \tau_i^{k'}(\mathbf{A}^{1/2}) \right) \\ &= O\left(\frac{\sqrt{nk} \log n}{\delta \cdot \epsilon^{2.5}} \right), \end{aligned} \tag{2.25}$$

where the last step follows from plugging in (2.24).

Fixing $\delta = \Theta(1)$, by applying an input sparsity time low-rank approximation algorithm to $\tilde{\mathbf{A}}$ (which by (2.25) has dimensions $\tilde{O}(\sqrt{nk}/\epsilon^{2.5}) \times \tilde{O}(\sqrt{nk}/\epsilon^{2.5})$ and so has just $\tilde{O}\left(\frac{nk}{\epsilon^5}\right)$ entries) we can find a near-optimal low-rank approximation of $\mathbf{AS}_1$, and thus for $\mathbf{A}$ in $n \cdot \text{poly}(k/\epsilon)$ time. However, in our final algorithm, we take a somewhat different approach. We are able to show that using appropriate sampling probabilities, we can in fact sample $\tilde{\mathbf{A}}$ that is a projection-cost-preserving sketch of $\mathbf{AS}_1$ for *spectral norm* error. As we will see, recovering a near-optimal spectral norm low-rank approximation to $\mathbf{AS}_1$ suffices to recover a near-optimal Frobenius norm approximation to $\mathbf{A}$, and allows us to improve ϵ dependencies in our final runtime. We first define this notation of projection-cost-preservation:

Definition 2.4.2 (Spectral Norm Projection-Cost-Preserving Sketch). $\mathbf{R} \in \mathbb{R}^{n' \times d}$ *is an* (ϵ, α, k)*-spectral PCP of* $\mathbf{C} \in \mathbb{R}^{n \times d}$ *if for all rank-*k *projection matrices* $\mathbf{P} \in \mathbb{R}^{d \times d}$*:*

$$(1-\epsilon)\|\mathbf{C} - \mathbf{CP}\|_2^2 - \alpha \leq \|\mathbf{R} - \mathbf{RP}\|_2^2 \leq (1+\epsilon)\|\mathbf{C} - \mathbf{CP}\|_F^2 + \alpha.$$

We then give show that sampling using the $k' = O(k/\epsilon)$ ridge leverage scores of $\mathbf{A}^{1/2}$ yield such a spectral PCP of $\mathbf{AS}_1$ with good probability.

Lemma 2.4.3 (Spectral Norm Row PCP). *For any PSD* $\mathbf{A} \in \mathbb{R}^{n \times n}$*, and* $\epsilon, \delta \in (0, 1/2]$ *let* $k' = \lceil ck/\epsilon^2 \rceil$ *and for all* $i \in [n]$ *let* $\tilde{\tau}_i^{k'}(\mathbf{A}^{1/2}) \geq \tau_i^{k'}(\mathbf{A}^{1/2})$ *be an overestimate for the* i^{th} *rank-*k' *ridge leverage score of* $\mathbf{A}^{1/2}$*. Let* $\tilde{\ell}_i = \sqrt{\frac{16n}{k'}} \cdot \tilde{\tau}_i^{k'}(\mathbf{A}^{1/2})$*,* $p_i = \frac{\tilde{\ell}_i}{\sum_i \tilde{\ell}_i}$*, and* $t \geq \frac{c' \log(n/\delta)}{\epsilon^2} \cdot \sum_i \tilde{\ell}_i$*. Construct weighted sampling matrices* $\mathbf{S}_1, \mathbf{S}_2 \in \mathbb{R}^{n \times t}$*, each whose* j^{th} *column is set to* $\frac{1}{\sqrt{tp_i}} \mathbf{e}_i$ *with probability* p_i*.*

For sufficiently large constants c, c'*, with probability* $\geq 1 - \delta$*,* $\tilde{\mathbf{A}} = \mathbf{S}_2^T \mathbf{AS}_1$ *is an* (ϵ, α, k)*-spectral PCP of* $\mathbf{AS}_1$ *with* $\alpha = \frac{\epsilon}{k}\|\mathbf{A} - \mathbf{A}_k\|_F^2$*.*

In Lemma 2.4.3, $k' = O(k/\epsilon^2)$, so again applying Lemma 2.2.3, the sum of rank-k' ridge leverage scores is bounded by

$$\sum_i \tau_i^{k'}(\mathbf{A}^{1/2}) = O(k/\epsilon^2). \tag{2.26}$$

Note that if each $\tilde{\tau}_i^{k'}(\mathbf{A}^{1/2})$ is a constant factor approximation to $\tau_i^{k'}(\mathbf{A}^{1/2})$, we can use sample size:

$$\begin{aligned} t &= \frac{c' \log(n/\delta)}{\epsilon^2} \cdot \sum_i \tilde{\ell}_i \\ &= \frac{c' \log(n/\delta)}{\epsilon^2} \cdot \sum_i \left(4\epsilon \sqrt{\frac{n}{k}} \cdot \tilde{\tau}_i^{k'}(\mathbf{A}^{1/2}) \right) \\ &= O\left(\frac{\log(n/\delta)}{\epsilon} \sqrt{\frac{n}{k}} \cdot \sum_i \tau_i^{k'}(\mathbf{A}^{1/2}) \right) \\ &= O\left(\frac{\sqrt{nk} \log(n/\delta)}{\epsilon^3} \right), \end{aligned}$$

where the last step follows from plugging in (2.26).

2.4.3 Spectral Norm Projection-Cost-Preservation

In this section we prove Lemma 2.4.3. In Section 2.4.4 we prove Lemma 2.4.1 using similar techniques. Both proofs are similar to proofs of projection-cost-preservation that we gave in [CEM+15] and [CMM17], e.g., the proof of Lemma 2.2.12. However, we must deal with some complications since we are not sampling using approximations to the actual ridge leverage scores of $\mathbf{AS}_1$.

Proof of Lemma 2.4.3. For conciseness write $\mathbf{C} \overset{\text{def}}{=} \mathbf{AS}_1$ and write the singular value decomposition $\mathbf{A} = \mathbf{U\Sigma U}^T$. We first show a claim that allows us to reduce proving the desired PCP property to showing a straightforward approximation bound between $\tilde{\mathbf{A}}$ and $\mathbf{C}$.

Claim 2.4.4. *If for all rank-k orthogonal projection matrices $\mathbf{P} \in \mathbb{R}^{n \times n}$,*

$$\|(\mathbf{I}-\mathbf{P})(\tilde{\mathbf{A}}^T\tilde{\mathbf{A}} - \mathbf{C}^T\mathbf{C})(\mathbf{I}-\mathbf{P})\|_2 \le \epsilon \|\mathbf{C}(\mathbf{I}-\mathbf{P})\|_2^2 + \frac{\epsilon}{k}\|\mathbf{A}-\mathbf{A}_k\|_F^2 \tag{2.27}$$

then $\tilde{\mathbf{A}}$ is an (ϵ, α, k)-spectral PCP of $\mathbf{C} = \mathbf{AS}_1$ with $\alpha = \frac{\epsilon}{k}\|\mathbf{A}-\mathbf{A}_k\|_F^2$.

Proof. Applying the triangle inequality we have:

$$\begin{aligned}\|\tilde{\mathbf{A}}(\mathbf{I}-\mathbf{P})\|_2^2 &= \|(\mathbf{I}-\mathbf{P})\tilde{\mathbf{A}}^T\tilde{\mathbf{A}}(\mathbf{I}-\mathbf{P})\|_2\\ &= \|(\mathbf{I}-\mathbf{P})[\mathbf{C}^T\mathbf{C}+(\tilde{\mathbf{A}}^T\tilde{\mathbf{A}}-\mathbf{C}^T\mathbf{C})](\mathbf{I}-\mathbf{P})\|_2\\ &\leq \|\mathbf{C}(\mathbf{I}-\mathbf{P})\|_2^2+\|(\mathbf{I}-\mathbf{P})(\tilde{\mathbf{A}}^T\tilde{\mathbf{A}}-\mathbf{C}^T\mathbf{C})(\mathbf{I}-\mathbf{P})\|_2.\end{aligned}$$

Symmetrically we have:

$$\|\tilde{\mathbf{A}}(\mathbf{I}-\mathbf{P})\|_2^2 \geq \|\mathbf{C}(\mathbf{I}-\mathbf{P})\|_2^2-\|(\mathbf{I}-\mathbf{P})(\tilde{\mathbf{A}}^T\tilde{\mathbf{A}}-\mathbf{C}^T\mathbf{C})(\mathbf{I}-\mathbf{P})\|_2.$$

Thus, if (2.27) holds,

$$\|\tilde{\mathbf{A}}(\mathbf{I}-\mathbf{P})\|_2^2 \leq (1+\epsilon)\|\mathbf{C}(\mathbf{I}-\mathbf{P})\|_2^2+\frac{\epsilon}{k}\|\mathbf{A}-\mathbf{A}_k\|_F^2 \tag{2.28}$$

and

$$\|\tilde{\mathbf{A}}(\mathbf{I}-\mathbf{P})\|_2^2 \geq (1-\epsilon)\|\mathbf{C}(\mathbf{I}-\mathbf{P})\|_2^2-\frac{\epsilon}{k}\|\mathbf{A}-\mathbf{A}_k\|_F^2. \tag{2.29}$$

Together, (2.28) and (2.29) give that $\tilde{\mathbf{A}}$ is an (ϵ,α,k)-spectral PCP of $\mathbf{C}=\mathbf{A}\mathbf{S}_1$ with $\alpha=\frac{\epsilon}{k}\|\mathbf{A}-\mathbf{A}_k\|_F^2$ by Definition 2.4.2, giving the claim. □

With Claim 2.4.4 in place, to prove the lemma, it suffices to show that (2.27) holds for all rank-k projection matrices with probability $\geq 1-\delta$. We will in fact show a stronger statement – that (2.27) holds for *all projection matrices*, with rank up to n. However, we do not need this stronger result in our final application of the lemma.

Let m be the largest index with $\sigma_m(\mathbf{A})^2 \geq \frac{\epsilon^2}{k}\|\mathbf{A}-\mathbf{A}_k\|_F^2$ (and let $m=0$ if no such singular value exists). Let $\mathbf{U}_H \in \mathbb{R}^{n\times m}$ contain the top m 'head' singular vectors of $\mathbf{A}$ (the first m columns of $\mathbf{U}$ and let $\mathbf{U}_T \in \mathbb{R}^{\text{rank}(\mathbf{A})-m}$ contain the remaining 'tail' singular vectors (the last $\text{rank}(\mathbf{A})-m$ columns of $\mathbf{U}$). Let $\mathbf{C}_H=\mathbf{U}_H\mathbf{U}_H^T\mathbf{C}$ and $\mathbf{C}_T=\mathbf{U}_T\mathbf{U}_T^T\mathbf{C}$. $\mathbf{C}_H$ is the projection of $\mathbf{C}$ onto $\mathbf{A}$'s top singular vectors. $\mathbf{C}_T$ is correspondingly the projection onto $\mathbf{A}$'s bottom singular vectors. Further, since the column span of $\mathbf{C}$ is a subset of that of $\mathbf{A}$ we have:

$$\mathbf{C}_H+\mathbf{C}_T=\mathbf{C}.$$

Applying the triangle inequality we can bound the left hand side of (2.27) by:

$$\begin{aligned}\|(\mathbf{I}-\mathbf{P})(\tilde{\mathbf{A}}^T\tilde{\mathbf{A}}-\mathbf{C}^T\mathbf{C})(\mathbf{I}-\mathbf{P})\|_2 \leq \|(\mathbf{I}{-}\mathbf{P})(\mathbf{C}_H^T\mathbf{S}_2\mathbf{S}_2^T\mathbf{C}_H-\mathbf{C}_H^T\mathbf{C}_H)(\mathbf{I}-\mathbf{P})\|_2 \\ +\|(\mathbf{I}-\mathbf{P})(\mathbf{C}_T^T\mathbf{S}_2\mathbf{S}_2^T\mathbf{C}_T-\mathbf{C}_T^T\mathbf{C}_T)(\mathbf{I}-\mathbf{P})\|_2 \\ +2\|(\mathbf{I}-\mathbf{P})\mathbf{C}_H^T\mathbf{S}_2\mathbf{S}_2^T\mathbf{C}_T(\mathbf{I}-\mathbf{P})\|_2. \qquad (2.30)\end{aligned}$$

We bound each of the terms in the above sum separately. Specifically we show:

Claim 2.4.5 (Head Term). *With probability $\geq 1-\delta/4$, for all orthogonal projection matrices $\mathbf{P}\in\mathbb{R}^{n\times n}$,*

$$\|(\mathbf{I}-\mathbf{P})(\mathbf{C}_H^T\mathbf{S}_2\mathbf{S}_2^T\mathbf{C}_H-\mathbf{C}_H^T\mathbf{C}_H)(\mathbf{I}-\mathbf{P})\|_2 \leq \epsilon\|\mathbf{C}(\mathbf{I}-\mathbf{P})\|_2^2.$$

Claim 2.4.6 (Tail Term). *With probability $\geq 1-\delta/4$, for all orthogonal projection matrices $\mathbf{P}\in\mathbb{R}^{n\times n}$,*

$$\|(\mathbf{I}-\mathbf{P})(\mathbf{C}_T^T\mathbf{S}_2\mathbf{S}_2^T\mathbf{C}_T-\mathbf{C}_T^T\mathbf{C}_T)(\mathbf{I}-\mathbf{P})\|_2 \leq \frac{13\epsilon^2}{k}\|\mathbf{A}-\mathbf{A}_k\|_F^2.$$

Claim 2.4.7 (Cross Term). *With probability $\geq 1-\delta/2$, for all orthogonal projection matrices $\mathbf{P}\in\mathbb{R}^{n\times n}$,*

$$\|(\mathbf{I}-\mathbf{P})\mathbf{C}_H^T\mathbf{S}_2\mathbf{S}_2^T\mathbf{C}_T(\mathbf{I}-\mathbf{P})\|_2 \leq \frac{10\epsilon}{k}\|\mathbf{A}-\mathbf{A}_k\|_F^2+4\epsilon\|\mathbf{C}(\mathbf{I}-\mathbf{P})\|_F^2.$$

Combining these three bounds, after adjusting constant factors on ϵ by making the constants c and c' in the rank parameter k' and sample size t large enough, will give that (2.27) holds with probability $\geq 1-\delta$ and thus the lemma via Claim 2.4.4. For the remainder of the proof we thus fix $c=1$ so $k'=\lceil k/\epsilon^2\rceil$.

Head Term:

We start by proving Claim 2.4.5. We first show that the ridge scores of $\mathbf{A}^{1/2}$ upper bound the row norms of $\mathbf{U}_H$ (which by Lemma 2.2.5 are equal to its leverage scores since it has orthonormal columns) and therefore the leverage scores of $\mathbf{C}_H$, since its columns lie within the column span of $\mathbf{U}_H$.

Claim 2.4.8. *For any p with $\sigma_p(\mathbf{A})^2\geq\frac{1}{k}\|\mathbf{A}-\mathbf{A}_k\|_F^2$:*

$$\sqrt{\frac{16n}{k}}\cdot\tau_i^k(\mathbf{A}^{1/2})\geq\|(\mathbf{U}_p)_i\|_2^2, \qquad (2.31)$$

where $\mathbf{U}_p \in \mathbb{R}^{n\times p}$ *has columns equal to the top* p *singular vectors of* $\mathbf{A}$ *(the first* p *columns of* $\mathbf{U}$*) and* $(\mathbf{U}_p)_i$ *is its* i^{th} *row.*

Proof. If $p = 0$ and so $\mathbf{U}_p$ contains no singular vectors, (2.31) is true vacuously since all row norms are 0 and all leverage scores are ≥ 0. Otherwise we write:

$$\begin{aligned}\tau_i^k(\mathbf{A}) &= \mathbf{a}_i^T\left(\mathbf{A}^2 + \frac{\|\mathbf{A}-\mathbf{A}_k\|_F^2}{k}I\right)^{-1}\mathbf{a}_i\\ &= \mathbf{e}_i^T\mathbf{U}\hat{\mathbf{\Sigma}}\mathbf{U}^T\mathbf{e}_i,\end{aligned}$$

where $\hat{\mathbf{\Sigma}}$ is a diagonal matrix with $\hat{\mathbf{\Sigma}}_{j,j} = \frac{\sigma_j(\mathbf{A})^2}{\sigma_j(\mathbf{A})^2 + \frac{\|\mathbf{A}-\mathbf{A}_k\|_F^2}{k}}$. We can then write:

$$\begin{aligned}\tau_i^k(\mathbf{A}) &= \sum_{j=1}^n \mathbf{U}_{i,j}^2 \cdot \hat{\mathbf{\Sigma}}_{j,j}\\ &\geq \sum_{j=1}^p \mathbf{U}_{i,j}^2 \cdot \hat{\mathbf{\Sigma}}_{j,j} && \text{(truncate sum)}\\ &\geq \sum_{j=1}^p \left(\mathbf{U}_{i,j}^2 \cdot \frac{\sigma_j(\mathbf{A})^2}{2\sigma_j(\mathbf{A})^2}\right) && \text{(By assumption, } \sigma_j(\mathbf{A})^2 \geq \sigma_p(\mathbf{A})^2 \geq \tfrac{1}{k}\|\mathbf{A}-\mathbf{A}_k\|_F^2\text{)}\\ &\geq \frac{1}{2}\sum_{j=1}^m \mathbf{U}_{i,j}^2 = \frac{1}{2}\|(\mathbf{U}_p)_i\|_2^2.\end{aligned}$$

This gives the claim since by Lemma 2.3.1 $\tau_i^k(\mathbf{A}) \leq 2\sqrt{\frac{n}{k}}\tau_i^k(\mathbf{A}^{1/2})$ so

$$\|(\mathbf{U}_p)_i\|_2^2 \leq 2\tau_i^k(\mathbf{A}) \leq 4\sqrt{\frac{n}{k}}\tau_i^k(\mathbf{A}^{1/2}).$$

□

With Claim 2.4.8 in place we can now prove our head term bound, Claim 2.4.5.

Proof of Claim 2.4.5. We apply Lemma 2.4.8 with $k' = \lceil k/\epsilon^2\rceil$. Recall that we chose m such that $\sigma_m(\mathbf{A})^2 \geq \frac{\epsilon^2}{k}\|\mathbf{A}-\mathbf{A}_k\|_F^2$ and so $\sigma_m^2(\mathbf{A}) \geq \frac{1}{k'}\|\mathbf{A}-\mathbf{A}_{k'}\|_F^2$. Additionally, we set in the statement of Lemma 2.4.3, $\tilde{\ell}_i = \sqrt{\frac{16n}{k'}} \cdot \tilde{\tau}_i^{k'}(\mathbf{A}^{1/2})$. So Claim 2.4.8 gives:

$$\tilde{\ell}_i = \sqrt{\frac{16n}{k'}} \cdot \tilde{\tau}_i^{k'}(\mathbf{A}^{1/2}) \geq \|(\mathbf{U}_m)_i\|_2^2 = \|(\mathbf{U}_H)_i\|_2^2. \tag{2.32}$$

As shown in Lemma 2.2.5, since $\mathbf{C}_H$ is spanned by the columns of $\mathbf{U}_H$, its row leverage scores are upper bounded by the row norms of $\mathbf{U}_H$. Formally:

Corollary 2.4.9. $\ell_i(\mathbf{C}_H) \leq \|(\mathbf{U}_H)_i\|_2^2$.

Corollary 2.4.9 combined with (2.32) gives $\tilde{\ell}_i \geq \ell_i(\mathbf{C}_H)$. Further, $t \geq \frac{c' \log(n/\delta)}{\epsilon^2} \cdot \sum_i \tilde{\ell}_i$ so if we set c' large enough, we have by the matrix Chernoff bound of Lemma 2.2.6, with probability $\geq 1 - \delta/4$:

$$(1-\epsilon)\mathbf{C}_H^T\mathbf{C}_H \preceq \mathbf{C}_H^T\mathbf{S}_2\mathbf{S}_2^T\mathbf{C}_H \prec (1+\epsilon)\mathbf{C}_H^T\mathbf{C}_H. \tag{2.33}$$

This in turn gives that for any $\mathbf{P} \in \mathbb{R}^{n \times n}$,

$$\begin{aligned}\|(\mathbf{I}-\mathbf{P})(\mathbf{C}_H^T\mathbf{S}_2\mathbf{S}_2^T\mathbf{C}_H - \mathbf{C}_H^T\mathbf{C}_H)(\mathbf{I}-\mathbf{P})\|_2 &\leq \epsilon\|(\mathbf{I}-\mathbf{P})\mathbf{C}_H^T\mathbf{C}_H(\mathbf{I}-\mathbf{P})\|_2 \\ &= \epsilon\|\mathbf{C}_H(\mathbf{I}-\mathbf{P})\|_2^2 \\ &\leq \epsilon\|\mathbf{C}(\mathbf{I}-\mathbf{P})\|_2^2, \end{aligned} \tag{2.34}$$

yielding Claim 2.4.5 □

Note that while $\mathbf{C}_H$ is a random variable that depends on the choice of the column sampling matrix $\mathbf{S}_1$, Claim 2.4.5 holds for any $\mathbf{S}_1$.

Tail Term:

We next prove the tail bound of Claim 2.4.6.

Proof of Claim 2.4.6. We can loosely bound via the triangle inequality and the fact that for any projection matrix $\mathbf{P}$, $\|\mathbf{I}-\mathbf{P}\|_2 \leq 1$:

$$\begin{aligned}\|(\mathbf{I}-\mathbf{P})(\mathbf{C}_T^T\mathbf{S}_2\mathbf{S}_2^T\mathbf{C}_T - \mathbf{C}_T^T\mathbf{C}_T)(\mathbf{I}-\mathbf{P})\|_2 &\leq \|\mathbf{C}_T^T\mathbf{S}_2\mathbf{S}_2^T\mathbf{C}_T - \mathbf{C}_T^T\mathbf{C}_T\|_2 \\ &\leq \|\mathbf{S}_2\mathbf{C}_T\|_2^2 + \|\mathbf{C}_T\|_2^2. \end{aligned} \tag{2.35}$$

Since $k' = \lceil k/\epsilon^2 \rceil$, recalling that we let $\mathbf{x}_i$ denote the i^{th} column of $\mathbf{A}^{1/2}$:

$$\begin{aligned}\tilde{\ell}_i = \sqrt{\frac{16n}{k'}} \cdot \tilde{\tau}_i^{k'}(\mathbf{A}^{1/2}) = \sqrt{\frac{16n}{k'}}\mathbf{x}_i^T\left(\mathbf{A} + \frac{\|\mathbf{A}^{1/2} - \mathbf{A}_{k'}^{1/2}\|_F^2}{k'}\right)^+ \mathbf{x}_i \\ \geq \mathbf{x}_i^T\left(\mathbf{A} + \frac{\|\mathbf{A}^{1/2} - \mathbf{A}_k^{1/2}\|_F^2}{\sqrt{nk'}}\right)^+ \mathbf{x}_i \\ = \tau_{i,\lambda}(\mathbf{A}^{1/2})\end{aligned}$$

for $\lambda = \frac{\|\mathbf{A}^{1/2} - \mathbf{A}_k^{1/2}\|_F^2}{\sqrt{nk'}} \leq \frac{\epsilon\|\mathbf{A}^{1/2} - \mathbf{A}_k^{1/2}\|_F^2}{\sqrt{nk}}$ by Definition 2.2.1 Thus for $\mathbf{S} = \mathbf{S}_1$ or $\mathbf{S} = \mathbf{S}_2$, for sufficiently large c' in our sample size $t \geq \frac{c' \log(n/\delta)}{\epsilon^2} \cdot \sum_i \tilde{\ell}_i$, by the ridge leverage score matrix Chernoff bound of Corollary 2.2.11, with probability $\geq 1-\delta/4$,

$$(1-\epsilon)\mathbf{A}^{1/2}\mathbf{S}\mathbf{S}^T\mathbf{A}^{1/2} - \frac{\epsilon\|\mathbf{A}^{1/2}-\mathbf{A}_k^{1/2}\|_F^2}{\sqrt{nk}} \preceq \mathbf{A} \preceq (1+\epsilon)\mathbf{A}^{1/2}\mathbf{S}\mathbf{S}^T\mathbf{A}^{1/2} + \frac{\epsilon\|\mathbf{A}^{1/2}-\mathbf{A}_k^{1/2}\|_F^2}{\sqrt{nk}}. \tag{2.36}$$

If (2.36) holds, which occurs with probability $\geq 1-\delta/4$, we have:

$$\sigma_1^2(\mathbf{S}^T\mathbf{A}^{1/2}(\mathbf{I}-\mathbf{U}_T\mathbf{U}_T^T)) \leq (1+\epsilon)\sigma_1^2(\mathbf{A}^{1/2}(\mathbf{I}-\mathbf{U}_T\mathbf{U}_T^T)) + \frac{\epsilon\|\mathbf{A}^{1/2}-\mathbf{A}_k^{1/2}\|_F^2}{\sqrt{nk}}$$

and so,

$$\begin{aligned}
\|\mathbf{S}_2^T\mathbf{C}_T\|_2^2 &= \|\mathbf{S}_2^T(\mathbf{I}-\mathbf{U}_T\mathbf{U}_T^T)\mathbf{A}\mathbf{S}_1\|_2^2 \\
&\leq \sigma_1^2(\mathbf{S}_2^T\mathbf{A}^{1/2}(\mathbf{I}-\mathbf{U}_T\mathbf{U}_T^T)) \cdot \sigma_1^2((\mathbf{I}-\mathbf{U}_T\mathbf{U}_T^T)\mathbf{A}^{1/2}\mathbf{S}_1) \\
&\leq 2(1+\epsilon)^2\sigma_1^4(\mathbf{A}^{1/2}(\mathbf{I}-\mathbf{U}_T\mathbf{U}_T)^T) + \frac{2\epsilon^2\|\mathbf{A}^{1/2}-\mathbf{A}_k^{1/2}\|_F^4}{nk} \\
&\leq 8\|\mathbf{A}(\mathbf{I}-\mathbf{U}_T\mathbf{U}_T^T)\|_2^2 + \frac{2\epsilon^2\|\mathbf{A}-\mathbf{A}_k\|_F^2}{k} \\
&\leq \frac{10\epsilon^2\|\mathbf{A}-\mathbf{A}_k\|_F^2}{k}.
\end{aligned} \tag{2.37}$$

The first bound is via submultiplicativity of the spectral norm. The second is via the arithmetic-geometric mean inequality. The third follows from the fact that $\|\mathbf{A}^{1/2}-\mathbf{A}_k^{1/2}\|_F^4 \leq n\|\mathbf{A}-\mathbf{A}_k\|_F^2$ by an ℓ_1/ℓ_2 bound and the last follows from the fact that $\|\mathbf{A}(\mathbf{I}-\mathbf{U}_T\mathbf{U}_T^T)\|_2^2 \leq \frac{\epsilon^2\|\mathbf{A}-\mathbf{A}_k\|_F^2}{k}$ by our choice of m and hence of $\mathbf{U}_H$ and $\mathbf{U}_T$. Similarly:

$$\begin{aligned}
\|\mathbf{C}_T\|_2^2 &= \sigma_1^2((\mathbf{I}-\mathbf{U}_T\mathbf{U}_T^T)\mathbf{A}\mathbf{S}_1) \\
&\leq \sigma_1^2(\mathbf{S}_1^T\mathbf{A}^{1/2}(\mathbf{I}-\mathbf{U}_T\mathbf{U}_T^T)) \cdot \sigma_1^2(\mathbf{A}^{1/2}(\mathbf{I}-\mathbf{U}_T\mathbf{U}_T^T)) \\
&\leq (1+\epsilon)\sigma_1^4(\mathbf{A}^{1/2}(\mathbf{I}-\mathbf{U}_T\mathbf{U}_T^T)) + \frac{\epsilon\|\mathbf{A}^{1/2}-\mathbf{A}_k^{1/2}\|_F^2}{\sqrt{nk}} \cdot \sigma_1^2(\mathbf{A}^{1/2}(\mathbf{I}-\mathbf{U}_T\mathbf{U}_T^T)) \\
&\leq \frac{3\epsilon^2\|\mathbf{A}-\mathbf{A}_k\|_F^2}{k}.
\end{aligned} \tag{2.38}$$

Overall, plugging (2.37) and (2.38) back into (2.35) gives, for every projection matrix $\mathbf{P} \in \mathbb{R}^{n\times n}$:

$$\|(\mathbf{I}-\mathbf{P})(\mathbf{C}_T^T\mathbf{S}_2\mathbf{S}_2^T\mathbf{C}_T - \mathbf{C}_T^T\mathbf{C}_T)(\mathbf{I}-\mathbf{P})\|_2 \leq \frac{13\epsilon^2}{k}\|\mathbf{A}-\mathbf{A}_k\|_F^2. \tag{2.39}$$

This bound holds as long as (2.36) does, which happens with probability $\geq 1-\delta/4$, yielding the claim. □

Cross Term:

We finally prove our cross term bound, Claim 2.4.7.

Proof of Claim 2.4.7. By submultiplicativity of the spectral norm:

$$\|(\mathbf{I}-\mathbf{P})\mathbf{C}_T^T\mathbf{S}_2\mathbf{S}_2^T\mathbf{C}_H(\mathbf{I}-\mathbf{P})\|_2 \le \|\mathbf{C}_T^T\mathbf{S}_2\|_2 \cdot \|\mathbf{S}_2^T\mathbf{C}_H(\mathbf{I}-\mathbf{P})\|_2. \tag{2.40}$$

As shown above for our tail bound, if (2.36) holds (which happens with probability $\ge 1-\delta/4$, $\|\mathbf{C}_T^T\mathbf{S}_2\|_2 \le \sqrt{\frac{10\epsilon^2}{k}}\|\mathbf{A}-\mathbf{A}_k\|_F$. Further, if (2.33) holds, which also happens with probability $\ge 1-\delta/4$, then for any $\mathbf{P} \in \mathbb{R}^{n\times n}$,

$$\|\mathbf{S}_2^T\mathbf{C}_H(\mathbf{I}-\mathbf{P})\|_2 \le (1+\epsilon)\|\mathbf{C}_H(\mathbf{I}-\mathbf{P})\|_2.$$

Plugging these bounds back into (2.40), with probability $\ge 1-\delta/2$:

$$\begin{aligned}\|(\mathbf{I}-\mathbf{P})\mathbf{C}_T^T\mathbf{S}_2\mathbf{S}_2^T\mathbf{C}_H(\mathbf{I}-\mathbf{P})\|_2 &\le \sqrt{\frac{10\epsilon^2}{k}}\|\mathbf{A}-\mathbf{A}_k\|_F \cdot (1+\epsilon)\|\mathbf{C}_H(\mathbf{I}-\mathbf{P})\|_2\\ &\le \frac{10\epsilon}{k}\|\mathbf{A}-\mathbf{A}_k\|_F^2 + 4\epsilon\|\mathbf{C}(\mathbf{I}-\mathbf{P})\|_2^2, \end{aligned}\tag{2.41}$$

where the last bound follows form the arithmetic-geometric mean inequality. □

Completing the proof of Lemma 2.4.3:

With Claims 2.4.5, 2.4.6, and 2.4.7 in place, by a union bound, with probability $\ge 1-\delta$, using the decomposition of (2.30) we have:

$$\begin{aligned}\|(\mathbf{I}-\mathbf{P})(\tilde{\mathbf{A}}^T\tilde{\mathbf{A}}-\mathbf{C}^T\mathbf{C})(\mathbf{I}-\mathbf{P})\|_2 &\le \|(\mathbf{I}-\mathbf{P})(\mathbf{C}_H^T\mathbf{S}_2\mathbf{S}_2^T\mathbf{C}_H-\mathbf{C}_H^T\mathbf{C}_H)(\mathbf{I}-\mathbf{P})\|_2\\ &\quad+\|(\mathbf{I}-\mathbf{P})(\mathbf{C}_T^T\mathbf{S}_2\mathbf{S}_2^T\mathbf{C}_T-\mathbf{C}_T^T\mathbf{C}_T)(\mathbf{I}-\mathbf{P})\|_2\\ &\quad+2\|(\mathbf{I}-\mathbf{P})\mathbf{C}_H^T\mathbf{S}_2\mathbf{S}_2^T\mathbf{C}_T(\mathbf{I}-\mathbf{P})\|_2\\ &\le \epsilon\|\mathbf{C}(\mathbf{I}-\mathbf{P})\|_2^2 + \frac{13\epsilon^2}{k}\|\mathbf{A}-\mathbf{A}_k\|_F^2 + \frac{20\epsilon}{k}\|\mathbf{A}-\mathbf{A}_k\|_F^2 + 8\epsilon\|\mathbf{C}(\mathbf{I}-\mathbf{P})\|_F^2\\ &\le 9\epsilon\|\mathbf{C}(\mathbf{I}-\mathbf{P})\|_2^2 + \frac{33\epsilon}{k}\|\mathbf{A}-\mathbf{A}_k\|_F^2.\end{aligned}$$

By Claim 2.4.4 this gives that $\tilde{\mathbf{A}}$ is a $(33\epsilon, \alpha, k)$-spectral PCP of $\mathbf{C} = \mathbf{A}\mathbf{S}_1$ with $\alpha = \frac{33\epsilon}{k}\|\mathbf{A}-\mathbf{A}_k\|_F^2$. This gives Lemma 2.4.3 after adjusting constants on ϵ by setting c, c' in the lemma statement sufficiently large.

□

2.4.4 Frobenius Norm Projection-Cost-Preservation

We now prove Lemma 2.4.1, showing that, with probability $\geq 1-\delta$, $\mathbf{AS}_1$ is an (ϵ, k)-PCP of $\mathbf{A}$ (Definition 2.1.3) and in turn $\mathbf{S}_2^T\mathbf{AS}_1$ is an $(\epsilon, 18/\delta, k)$-PCP of $\mathbf{AS}_1$ (Definition 2.1.4). The proof is quite similar to that of Lemma 2.4.3.

Proof of Lemma 2.4.1. Again denote $\mathbf{C} = \mathbf{AS}_1$ and write the singular value decomposition $\mathbf{A} = \mathbf{U\Sigma U}^T$. Let m be the largest index with $\sigma_m(\mathbf{A})^2 \geq \frac{\epsilon}{k}\|\mathbf{A}-\mathbf{A}_k\|_F^2$. Let $\mathbf{U}_H \in \mathbb{R}^{n\times m}$ contain the top m singular of $\mathbf{A}$ (the first m columns of $\mathbf{U}$ and let $\mathbf{U}_T \in \mathbb{R}^{n\times \text{rank}(\mathbf{A})-m}$ contain the remaining 'tail' singular vectors. Let $\mathbf{C}_H = \mathbf{U}_H\mathbf{U}_H^T\mathbf{C}$ and $\mathbf{C}_T = \mathbf{U}_T\mathbf{U}_T^T\mathbf{C}$ and note that $\mathbf{C}_H + \mathbf{C}_T = \mathbf{C}$. Again, throughout the proof we fix $c = 1$ so $k' = \lceil k/\epsilon\rceil$. We prove the result up to constant factors on ϵ, which we can then remove by setting c, c' large enough.

We first claim that $\mathbf{AS}_1$ is an (ϵ, k)-PCP of $\mathbf{A}$.

Claim 2.4.10. *With probability $\geq 1-\delta/8$, $\mathbf{C} = \mathbf{AS}_1$ is an (ϵ, k)-PCP of $\mathbf{A}$ (Definition 2.1.3).*

Proof. $\tilde{\ell}_i = \sqrt{\frac{16n}{k'}}\tilde{\tau}_i^{k'}(\mathbf{A}^{1/2})$. By the assumption that $\tilde{\tau}_i^{k'}(\mathbf{A}^{1/2}) \geq \tau_i^{k'}(\mathbf{A}^{1/2})$ for all i and Lemma 2.3.1 this gives, for all i

$$\tilde{\ell}_i \geq \sqrt{\frac{16n}{k'}}\tau_i^{k'}(\mathbf{A}^{1/2}) \geq \tau_i^{k'}(\mathbf{A}) \geq \tau_i^{k}(\mathbf{A}).$$

By Lemma 2.2.12, since $\mathbf{S}_1$ samples $t \geq \frac{c'\log n}{\delta\cdot\epsilon^2}\sum_i \tilde{\ell}_i$ columns, if c' is large enough, with probability $\geq 1-\delta/8$, $\mathbf{C} = \mathbf{AS}_1$ is an (ϵ, k)-PCP of $\mathbf{A}$. □

We also use the following bound, which holds if $\mathbf{C}$ is an (ϵ, k)-PCP of $\mathbf{A}$.

Claim 2.4.11. *If $\mathbf{C}$ is an (ϵ, k)-PCP of $\mathbf{A}$ then:*

1. $\|\mathbf{A}-\mathbf{A}_k\|_F^2 \leq (1+2\epsilon)\|\mathbf{C}-\mathbf{C}_k\|_F^2$.

2. $\|\mathbf{C}_T\|_F^2 \leq (1+7\epsilon)\|\mathbf{C}-\mathbf{C}_k\|_F^2$.

Proof. We prove the two conclusions of the claim separately.

Conclusion 1: Letting $\mathbf{Z} \in \mathbb{R}^{n\times k}$ contain as its columns $\mathbf{C}$'s top k singular vectors, we have from Definition 2.1.3:

$$\begin{aligned}\|(\mathbf{I}-\mathbf{ZZ}^T)\mathbf{A}\|_F^2 &\leq \frac{1}{1-\epsilon}\|(\mathbf{I}-\mathbf{ZZ}^T)\mathbf{C}\|_F^2\\ &\leq (1+2\epsilon)\|\mathbf{C}-\mathbf{C}_k\|_F^2\end{aligned}$$

for $\epsilon \in (0, 1/2]$. The conclusion follows since $\|\mathbf{A} - \mathbf{A}_k\|_F^2 \leq \|(\mathbf{I} - \mathbf{Z}\mathbf{Z}^T)\mathbf{A}\|_F^2$ since $\mathbf{A}_k$ is the best rank-k approximation to $\mathbf{A}$.

Conclusion 2: Recall that m is set to the largest index with $\sigma_m(\mathbf{A})^2 \geq \frac{\epsilon}{k}\|\mathbf{A} - \mathbf{A}_k\|_F^2$. Consider first the case in which $m \geq k$. In this case:

$$\|\mathbf{A} - \mathbf{U}_H\mathbf{U}_H^T\mathbf{A}\|_F^2 = \|\mathbf{A} - \mathbf{A}_m\|_F^2 \leq \|\mathbf{A} - \mathbf{A}_k\|_F^2. \tag{2.42}$$

Next consider the case in which $m \leq k$. In this case since, by definition, $\sigma_m(\mathbf{A})^2 \leq \frac{\epsilon}{k}\|\mathbf{A} - \mathbf{A}_k\|_F^2$,

$$\begin{aligned}\|\mathbf{A} - \mathbf{U}_H\mathbf{U}_H^T\mathbf{A}\|_F^2 = \|\mathbf{A} - \mathbf{A}_m\|_F^2 &\leq \|\mathbf{A} - \mathbf{A}_k\|_F^2 + (k-m)\sigma_m(\mathbf{A})^2 \\ &\leq (1+\epsilon)\|\mathbf{A} - \mathbf{A}_k\|_F^2. \end{aligned} \tag{2.43}$$

Using Definition 2.1.3 we have since $\mathbf{U}_H\mathbf{U}_H^T$ is a rank-k projection matrix:

$$\begin{aligned}\|\mathbf{C}_T\|_F^2 = \|\mathbf{C} - \mathbf{C}_H\|_F^2 = \|\mathbf{C} - \mathbf{U}_H\mathbf{U}_H^T\mathbf{C}\|_F^2 \\ &\leq (1+\epsilon)\|\mathbf{A} - \mathbf{U}_H\mathbf{U}_H^T\mathbf{A}\|_F^2 \\ &\leq (1+\epsilon)^2\|\mathbf{A} - \mathbf{A}_k\|_F^2. \end{aligned} \tag{2.44}$$

where the last bound follows from (2.42) and (2.43) Let $\mathbf{Z} \in \mathbb{R}^{n\times k}$ contain as its columns $\mathbf{C}$'s top k singular vectors. Again using Definition 2.1.3 and (2.44) we have:

$$\begin{aligned}\|\mathbf{C}_T\|_F^2 &\leq (1+\epsilon)\|\mathbf{A} - \mathbf{A}_k\|_F^2 \\ &\leq (1+\epsilon)^2\|(\mathbf{I} - \mathbf{Z}\mathbf{Z}^T)\mathbf{A}\|_F^2 \\ &\leq \frac{(1+\epsilon)^2}{1-\epsilon}\|(\mathbf{I} - \mathbf{Z}\mathbf{Z}^T)\mathbf{C}\|_F^2 \\ &\leq (1+7\epsilon)\|\mathbf{C} - \mathbf{C}_k\|_F^2 \end{aligned} \tag{2.45}$$

for $\epsilon \in (0, 1/2]$. This completes the claim. □

We next show that $\mathbf{S}_2^T\mathbf{A}\mathbf{S}_1$ is a PCP for $\mathbf{C} = \mathbf{A}\mathbf{S}_1$ with high probability. By the Pythagorean theorem:

$$\|\mathbf{C}(\mathbf{I} - \mathbf{P})\|_F^2 = \|\mathbf{C}_H(\mathbf{I} - \mathbf{P})\|_F^2 + \|\mathbf{C}_T(\mathbf{I} - \mathbf{P})\|_F^2.$$

Expanding using the identity $\|\mathbf{M}\|_F^2 = \operatorname{tr}(\mathbf{M}^T\mathbf{M})$ we can write

$$\begin{aligned}\|\tilde{\mathbf{A}}(\mathbf{I}-\mathbf{P})\|_F^2 = \|\mathbf{S}_2^T\mathbf{C}_H(\mathbf{I}-\mathbf{P})\|_F^2 + \|\mathbf{S}_2^T\mathbf{C}_T(\mathbf{I}-\mathbf{P})\|_F^2 \\ + 2\operatorname{tr}\left((\mathbf{I}-\mathbf{P})\mathbf{C}_H^T\mathbf{S}_2\mathbf{S}_2^T\mathbf{C}_T(\mathbf{I}-\mathbf{P})\right).\end{aligned} \tag{2.46}$$

As in the proof of Lemma 2.4.3 we bound each of the terms in (2.46) separately. Specifically we show:

Claim 2.4.12 (Head Term)**.** *With probability $\geq 1-\delta/8$, for all rank-k orthogonal projection matrices $\mathbf{P} \in \mathbb{R}^{n\times n}$,*

$$(1-\epsilon)\|\mathbf{C}_H(\mathbf{I}-\mathbf{P})\|_F^2 \leq \|\mathbf{S}_2^T\mathbf{C}_H(\mathbf{I}-\mathbf{P})\|_F^2 \leq (1+\epsilon)\|\mathbf{C}_H(\mathbf{I}-\mathbf{P})\|_F^2.$$

Claim 2.4.13 (Tail Term)**.** *With probability $\geq 1-\delta/2$, there is some fixed E with $|E| \leq 18/\delta \cdot \|\mathbf{C}-\mathbf{C}_k\|_F^2$ such that, for all rank-k projection matrices $\mathbf{P} \in \mathbb{R}^{n\times n}$,*

$$\|\mathbf{C}_T(\mathbf{I}-\mathbf{P})\|_F^2 - 40\epsilon\|\mathbf{C}-\mathbf{C}_k\|_F^2 \leq \|\mathbf{S}_2^T\mathbf{C}_T(\mathbf{I}-\mathbf{P})\|_F^2 + E \leq \|\mathbf{C}_T(\mathbf{I}-\mathbf{P})\|_F^2 + 40\epsilon\|\mathbf{C}-\mathbf{C}_k\|_F^2.$$

Claim 2.4.14 (Cross Term)**.** *With probability $\geq 1-\delta/4$, for all rank-k orthogonal projection matrices $\mathbf{P} \in \mathbb{R}^{n\times n}$,*

$$|\operatorname{tr}\left((\mathbf{I}-\mathbf{P})\mathbf{C}_H^T\mathbf{S}_2\mathbf{S}_2^T\mathbf{C}_T(\mathbf{I}-\mathbf{P})\right)| \leq 5\epsilon\|\mathbf{C}(\mathbf{I}-\mathbf{P})\|_F^2.$$

Head Term:

Proof of Claim 2.4.12. By Claim 2.4.8 applied to $k' = \lceil k/\epsilon \rceil$, since by definition $\sigma_m(\mathbf{A})^2 \geq \frac{\epsilon}{k}\|\mathbf{A}-\mathbf{A}_k\|_F^2 \geq \frac{1}{k'}\|\mathbf{A}-\mathbf{A}_{k'}\|_F^2$:

$$\begin{aligned}\tilde{\ell}_i &= \sqrt{\frac{16n\epsilon}{k}} \cdot \tilde{\tau}_i^{k'}(\mathbf{A}^{1/2}) \\ &\geq \sqrt{\frac{16n}{k'}} \cdot \tau_i^{k'}(\mathbf{A}^{1/2}) \\ &\geq \|(\mathbf{U}_H)_i\|_2^2.\end{aligned} \tag{2.47}$$

Further, by Corollary 2.4.9 since $\mathbf{U}_H$ is an orthonormal matrix that spans the columns of $\mathbf{C}_H$ we have $\ell_i(\mathbf{C}_H) \leq \|(\mathbf{U}_H)_i\|_2^2$ for all i. Thus $\tilde{\ell}_i \geq \ell_i(\mathbf{C}_H)$ for all i. Since $t \geq \frac{c' \log n}{\delta \cdot \epsilon^2} \cdot \sum_i \tilde{\ell}_i$, if we set c' large enough, by the matrix Chernoff bound of Lemma

2.2.6, with probability $\geq 1 - \delta/8$:

$$(1-\epsilon)\mathbf{C}_H^T\mathbf{C}_H \preceq \mathbf{C}_H^T\mathbf{S}_2\mathbf{S}_2^T\mathbf{C}_H \prec (1+\epsilon)\mathbf{C}_H^T\mathbf{C}_H.$$

This gives, for every $\mathbf{P} \in \mathbb{R}^{n\times n}$:

$$(1-\epsilon)\|\mathbf{C}_H(\mathbf{I}-\mathbf{P})\|_F^2 \leq \|\mathbf{S}_2^T\mathbf{C}_H(\mathbf{I}-\mathbf{P})\|_F^2 \leq (1+\epsilon)\|\mathbf{C}_H(\mathbf{I}-\mathbf{P})\|_F^2,$$

yielding the claim. □

Tail Term:

Proof of Claim 2.4.13. To prove the claim we need to show that, with probability $\geq 1-\delta/2$, there is some fixed E with $|E| \leq 18/\delta \cdot \|\mathbf{C}-\mathbf{C}_k\|_F^2$ such that, for all rank-k orthogonal projection matrices $\mathbf{P} \in \mathbb{R}^{n\times n}$,

$$\|\mathbf{C}_T(\mathbf{I}-\mathbf{P})\|_F^2 - 40\epsilon\|\mathbf{C}-\mathbf{C}_k\|_F^2 \leq \|\mathbf{S}_2^T\mathbf{C}_T(\mathbf{I}-\mathbf{P})\|_F^2 + E \leq \|\mathbf{C}_T(\mathbf{I}-\mathbf{P})\|_F^2 + 40\epsilon\|\mathbf{C}-\mathbf{C}_k\|_F^2.$$

We again split using Pythagorean theorem,

$$\|\mathbf{S}_2^T\mathbf{C}_T(\mathbf{I}-\mathbf{P})\|_F^2 = \|\mathbf{S}_2^T\mathbf{C}_T\|_F^2 - \|\mathbf{S}_2^T\mathbf{C}_T\mathbf{P}\|_F^2. \tag{2.48}$$

We set $E = \|\mathbf{C}_T\|_F^2 - \|\mathbf{S}_2^T\mathbf{C}_T\|_F^2$. We have

$$\mathbb{E}[\|\mathbf{S}_2\mathbf{C}_T\|_F^2] = \|\mathbf{C}_T\|_F^2 \leq (1+7\epsilon)\|\mathbf{C}-\mathbf{C}_k\|_F^2,$$

where the last bound follows from Claim 2.4.11 if $\mathbf{C}$ is an (ϵ, k)-PCP for $\mathbf{A}$. Thus by a Markov bound, with probability $\geq 1-\delta/4$,

$$\begin{aligned}|E| &\leq \frac{4}{\delta}\cdot(1+7\epsilon)\|\mathbf{C}-\mathbf{C}_k\|_F^2\\ &\leq \frac{18}{\delta}\|\mathbf{C}-\mathbf{C}_k\|_F^2\end{aligned}$$

since $\epsilon \in (0, 1/2]$. Additionally, we have, with probability $\geq 1-\delta/8$,

$$\|\mathbf{C}_T\|_2^2 \leq \frac{10\epsilon}{k}\|\mathbf{A}-\mathbf{A}_k\|_F^2 \text{ and } \|\mathbf{S}_2^T\mathbf{C}_T\|_2^2 \leq \frac{10\epsilon}{k}\|\mathbf{A}-\mathbf{A}_k\|_F^2$$

by an identical argument to that used for Claim 2.4.6. This gives, for every rank-k

projection $\mathbf{P}$:

$$\begin{aligned}\left|\|\mathbf{S}_2^T\mathbf{C}_T\mathbf{P}\|_F^2 - \|\mathbf{C}_T\mathbf{P}\|_F^2\right| &\le k(\|\mathbf{S}_2^T\mathbf{C}_T\|_2^2 + \|\mathbf{C}_T\|_2^2)\\ &\le 20\epsilon\|\mathbf{A}-\mathbf{A}_k\|_F^2\\ &\le 40\epsilon\|\mathbf{C}-\mathbf{C}_k\|_F^2, \end{aligned}\tag{2.49}$$

where the last bound follows from conclusion (1) of Claim 2.4.11 if $\mathbf{C}$ is an (ϵ, k)-PCP of $\mathbf{A}$. Overall, union bounding over the probability that $|E| \le \frac{18}{\delta}\|\mathbf{C}-\mathbf{C}_k\|_F^2$, that $\mathbf{C}$ is an (ϵ, k)-PCP for $\mathbf{A}$, and that (2.49), we have with probability $\ge 1-\delta/2$, for all rank-k projection matrices $\mathbf{P}$:

$$\begin{aligned}\|\mathbf{S}_2^T\mathbf{C}_T(\mathbf{I}-\mathbf{P})\|_F^2 &= \|\mathbf{S}_2^T\mathbf{C}_T\|_F^2 - \|\mathbf{S}_2^T\mathbf{C}_T\mathbf{P}\|_F^2\\ &= \|\mathbf{C}_T\|_F^2 - E - \|\mathbf{C}_T\mathbf{P}\|_F^2 - (\|\mathbf{S}_2^T\mathbf{C}_T\mathbf{P}\|_F^2 - \|\mathbf{C}_T\mathbf{P}\|_F^2)\\ &= \|\mathbf{C}_T(\mathbf{I}-\mathbf{P})\|_F^2 - E - (\|\mathbf{S}_2^T\mathbf{C}_T\mathbf{P}\|_F^2 - \|\mathbf{C}_T\mathbf{P}\|_F^2).\end{aligned}$$

Applying (2.49) then yields the claim. □

Cross Term:

Proof of Claim 2.4.14. We want to show that, with probability $\ge 1-\delta/4$, for all rank-k orthogonal projection matrices $\mathbf{P} \in \mathbb{R}^{n\times n}$,

$$|\operatorname{tr}\left((\mathbf{I}-\mathbf{P})\mathbf{C}_H^T\mathbf{S}_2\mathbf{S}_2^T\mathbf{C}_T(\mathbf{I}-\mathbf{P})\right)| \le 2\epsilon\|\mathbf{C}(\mathbf{I}-\mathbf{P})\|_F^2. \tag{2.50}$$

We can write:

$$\begin{aligned}\left|\operatorname{tr}\left((\mathbf{I}-\mathbf{P})\mathbf{C}_T^T\mathbf{S}_2\mathbf{S}_2^T\mathbf{C}_H(\mathbf{I}-\mathbf{P})\right)\right| &= \left|\operatorname{tr}\left(\mathbf{C}_T^T\mathbf{S}_2\mathbf{S}_2^T\mathbf{C}_H(\mathbf{I}-\mathbf{P})\right)\right|\\ &= \left|\operatorname{tr}\left(\mathbf{C}_T^T\mathbf{S}_2\mathbf{S}_2^T\mathbf{C}_H(\mathbf{C}^T\mathbf{C})^+(\mathbf{C}^T\mathbf{C})(\mathbf{I}-\mathbf{P})\right)\right|,\end{aligned}$$

where the first step follows from the cyclic property of trace and the fact that $(\mathbf{I}-\mathbf{P}) = (\mathbf{I}-\mathbf{P})^2$ since $\mathbf{P}$ is a projection matrix. The second step follows from inserting $(\mathbf{C}^T\mathbf{C})^+(\mathbf{C}^T\mathbf{C})$, which is the projection onto the row span of $\mathbf{C}$ has no effect as the rows of $\mathbf{C}_H = \mathbf{U}_H\mathbf{U}_H^T\mathbf{C}$ already lie in this span. $\langle\mathbf{M},\mathbf{N}\rangle = \operatorname{tr}(\mathbf{M}(\mathbf{C}^T\mathbf{C})^+\mathbf{N}^T)$ is a semi-inner product since $\mathbf{C}^T\mathbf{C}$ is positive semidefinite, so by Cauchy-Schwarz:

$$\left|\operatorname{tr}\left((\mathbf{I}-\mathbf{P})\mathbf{C}_T^T\mathbf{S}_2\mathbf{S}_2^T\mathbf{C}_H(\mathbf{I}-\mathbf{P})\right)\right| \le \|\mathbf{C}_T^T\mathbf{S}_2\mathbf{S}_2^T\mathbf{C}_H(\mathbf{C}^T\mathbf{C})^{+/2}\|_F \cdot \|\mathbf{C}(\mathbf{I}-\mathbf{P})\|_F. \tag{2.51}$$

Using the singular value decomposition $\mathbf{C} = \mathbf{X}\mathbf{S}\mathbf{Y}^T$, we can rewrite (2.51) as:

$$\begin{aligned}\left|\operatorname{tr}\left(\mathbf{C}_T^T\mathbf{S}_2\mathbf{S}_2^T\mathbf{C}_H(\mathbf{I}-\mathbf{P})\right)\right| &\leq \|\mathbf{C}_T^T\mathbf{S}_2\mathbf{S}_2^T\mathbf{U}_H\mathbf{U}_H^T\mathbf{X}\|_F \cdot \|\mathbf{C}(\mathbf{I}-\mathbf{P})\|_F \\ &\leq \|\mathbf{C}_T^T\mathbf{S}_2\mathbf{S}_2^T\mathbf{U}_H\|_F \cdot \|\mathbf{C}(\mathbf{I}-\mathbf{P})\|_F. \qquad (2.52)\end{aligned}$$

As shown in (2.47) in the proof of Claim 2.4.12, $\tilde{\ell}_i \geq \|(\mathbf{U}_H)_i\|_2^2$. Further, $t \geq \frac{c'}{\delta\epsilon^2}$. If c' is large enough, then applying the approximate matrix multiplication result of Lemma 2.2.16, with probability $\geq 1-\delta/8$:

$$\|\mathbf{C}_T^T\mathbf{S}_2\mathbf{S}_2^T\mathbf{U}_H\|_F \leq \epsilon\|\mathbf{C}_T\|_F.$$

Plugging this bound into (2.51) and (2.52) we have, with probability $\geq 1-\delta/8$,

$$\left|\operatorname{tr}\left((\mathbf{I}-\mathbf{P})\mathbf{C}_T^T\mathbf{S}_2\mathbf{S}_2^T\mathbf{C}_H(\mathbf{I}-\mathbf{P})\right)\right| \leq \epsilon\|\mathbf{C}_T\|_F \cdot \|\mathbf{C}(\mathbf{I}-\mathbf{P})\|_F.$$

Further, since by Claim 2.4.10, with probability $\geq 1-\delta/8$, $\mathbf{C} = \mathbf{A}\mathbf{S}_1$ is an (ϵ,k)-PCP of $\mathbf{A}$, using conclusion (2) of Claim 2.4.11, for any rank-k $\mathbf{P}$,

$$\begin{aligned}\|\mathbf{C}_T\|_F &\leq (1+7\epsilon)\|\mathbf{C}-\mathbf{C}_k\|_F \\ &\leq (1+7\epsilon)\|\mathbf{C}(\mathbf{I}-\mathbf{P})\|_F.\end{aligned}$$

Via a union bound, we thus have that with probability $\geq 1-\delta/4$:

$$\begin{aligned}\left|\operatorname{tr}\left((\mathbf{I}-\mathbf{P})\mathbf{C}_T^T\mathbf{S}_2\mathbf{S}_2^T\mathbf{C}_H(\mathbf{I}-\mathbf{P})\right)\right| &\leq \epsilon(1+7\epsilon)\|\mathbf{C}(\mathbf{I}-\mathbf{P})\|_F^2 \\ &\leq 5\epsilon\|\mathbf{C}(\mathbf{I}-\mathbf{P})\|_F^2,\end{aligned}$$

for $\epsilon \in (0, 1/2]$, which completes the claim. □

Completing the Proof of Lemma 2.4.1:

We now complete the lemma using Claims 2.4.10, 2.4.12, 2.4.13, and 2.4.14. Via a union bound, with probability $\geq 1-\delta$, the bounds in all claims hold. We thus have using (2.46), for E with $|E| \leq 18/\delta\|\mathbf{C}-\mathbf{C}_k\|_F^2$:

$$\begin{aligned}\|\tilde{\mathbf{A}}(\mathbf{I}-\mathbf{P})\|_F^2 &= \|\mathbf{S}_2^T\mathbf{C}_H(\mathbf{I}-\mathbf{P})\|_F^2 + \|\mathbf{S}_2^T\mathbf{C}_T(\mathbf{I}-\mathbf{P})\|_F^2 \\ &\quad + 2\operatorname{tr}\left((\mathbf{I}-\mathbf{P})\mathbf{C}_H^T\mathbf{S}_2\mathbf{S}_2^T\mathbf{C}_T(\mathbf{I}-\mathbf{P})\right) \\ &\leq (1+\epsilon)\|\mathbf{C}_H(\mathbf{I}-\mathbf{P})\|_F^2 + \|\mathbf{C}_T(\mathbf{I}-\mathbf{P})\|_F^2 + E + 40\epsilon\|\mathbf{C}-\mathbf{C}_k\|_F^2 + 10\epsilon\|\mathbf{C}(\mathbf{I}-\mathbf{P})\|_F^2 \\ &\leq (1+51\epsilon)\|\mathbf{C}(\mathbf{I}-\mathbf{P})\|_F^2 + E.\end{aligned}$$

Similarly,

$$\begin{aligned}\|\tilde{\mathbf{A}}(\mathbf{I}-\mathbf{P})\|_F^2 &\geq (1-\epsilon)\|\mathbf{C}_H(\mathbf{I}-\mathbf{P})\|_F^2 + \|\mathbf{C}_T(\mathbf{I}-\mathbf{P})\|_F^2 + E - 40\epsilon\|\mathbf{C}-\mathbf{C}_k\|_F^2 - 10\epsilon\|\mathbf{C}(\mathbf{I}-\mathbf{P})\|_F^2\\ &\geq (1-51\epsilon)\|\mathbf{C}(\mathbf{I}-\mathbf{P})\|_F^2 + E.\end{aligned}$$

This gives the lemma by adjusting constants on ϵ by making c and c' large enough. □

2.5 Full Low-Rank Approximation Algorithm

We are finally ready to give our main algorithm for relative error low-rank approximation of PSD matrices in $\tilde{O}(n\,\mathrm{poly}(k/\epsilon))$ time, Algorithm 1. In Section 2.5.1 we give pseudocode for and an analysis of the basic algorithm. In Section 2.5.2 we show how this algorithm can be modified to satisfy the stronger requirement that the outputted low-rank approximation is itself PSD. This requirement is often desired in applications, such as in kernel methods in machine learning [DM05, GM13].

2.5.1 Basic Algorithm

For constants c, c', we set $k_1 \overset{\text{def}}{=} \lceil ck/\epsilon \rceil$ and estimate the both the rank-k and rank-$c'k_1$ ridge leverage scores of $\mathbf{A}^{1/2}$ using the algorithm of Lemma 2.2.17 (Step 1). If c, c' are sufficiently large, sampling by the sum of these scores (Steps 2-3) ensures that $\mathbf{AS}_1$ is an (ϵ, k)-PCP for $\mathbf{A}$ and simultaneously, by applying Lemmas 2.4.1 and 2.4.3 that $\tilde{\mathbf{A}} = \mathbf{S}_2^T\mathbf{AS}_1$ is a row PCP in both spectral and Frobenius norm with rank k_1 and error $\epsilon = 1/2$ for $\mathbf{AS}_1$.

In conjunction, these guarantees ensure that we can apply an input sparsity time algorithm to $\tilde{\mathbf{A}}$ (Step 4) to find a rank-k_1 $\mathbf{Z} \in \mathbb{R}^{d\times k_1}$ satisfying:

$$\begin{aligned}\|\mathbf{AS_1} - \mathbf{AS}_1\mathbf{ZZ}^T\|_2^2 &= O\left(\|\mathbf{AS}_1 - (\mathbf{AS}_1)_{k_1}\|_2^2 + \frac{1}{k_1}\|\mathbf{A}-\mathbf{A}_k\|_F^2\right)\\ &= O\left(\frac{\epsilon}{k}\|\mathbf{A}-\mathbf{A}_k\|_F^2\right),\end{aligned}$$

where the final bound holds since $k_1 = \Theta(k/\epsilon)$. It is not hard to show that, due to this strong spectral norm bound, projecting $\mathbf{AS}_1$ to $\mathbf{Z}$ and taking the best rank-k approximation in the span gives a near-optimal Frobenius norm low-rank approximation to $\mathbf{AS}_1$ and hence $\mathbf{A}$ since $\mathbf{AS}_1$ is an (ϵ, k)-PCP of $\mathbf{A}$.

We still cannot afford to read $\mathbf{AS}_1$ in its entirety, so we employ a number of standard leverage score sampling techniques to perform this projection approximately.

In Step 5, we sample $\tilde{O}(k/\epsilon^2)$ columns of $\mathbf{AS}_1$ using the leverage scores of $\mathbf{Z}$ (its row norms since it is an orthonormal matrix) to form $\mathbf{AS}_1\mathbf{S}_3$. We argue that there is a good rank-k approximation to $\mathbf{AS}_1$ lying in both the column span of $\mathbf{AS}_1\mathbf{S}_3$ and the row span of $\mathbf{Z}^T$. In Step 6 we find a near-optimal such approximation by further sampling $\tilde{O}(k/\epsilon^4)$ rows $\mathbf{AS}_1$ by the leverage scores of $\mathbf{AS}_1\mathbf{S}_3$ (the row norms of $\mathbf{V}$, an orthonormal basis for its span), and computing the best rank-k approximation to the sampled matrix falling in the column span of $\mathbf{AS}_1\mathbf{S}_3$ and the row span of $\mathbf{Z}^T$.

Finally, in Step 7 we approximately project $\mathbf{A}$ itself to the span of this rank-k approximation by first sampling by the leverage scores of the approximation (the row norms of $\mathbf{M}$) and projecting.

The main result of this section bounds the runtime and approximation guarantees of Algorithm 1:

Theorem 2.5.1 (Sublinear Time Low-Rank Approximation)**.** *Given any PSD* $\mathbf{A} \in \mathbb{R}^{n\times n}$*, for sufficiently large constants* $c, c', c_1, c_2, c_3, c_4, c_5$*, for any* $k \in \mathbb{Z}^{\geq 1}$ *and* $\epsilon, \delta \in (0, 1/2]$*, Algorithm 1 accesses* $O\left(\frac{n\cdot k\log^2(n)}{\delta^2\cdot\epsilon^{2.5}} + \sqrt{n}k^{1.5}\cdot\log n\cdot\log k\cdot \text{poly}(1/\epsilon, 1/\delta)\right)$ *entries of* $\mathbf{A}$*, runs in*

$$\tilde{O}\left(\frac{nk^{\bar{\omega}-1}}{\delta^2\cdot\epsilon^{2(\bar{\omega}-1)}} + \sqrt{n}k^{\bar{\omega}-.5}\cdot\text{poly}(1/\epsilon, 1/\delta)\right)$$

time, and with probability $\geq 1-\delta$ *outputs* $M, N \in \mathbb{R}^{n\times k}$ *with:*

$$\|\mathbf{A} - \mathbf{MN}^T\|_F^2 \leq (1+\epsilon)\|\mathbf{A} - \mathbf{A}_k\|_F^2.$$

Algorithm 1 PSD Low-Rank Approximation

1. Let $k_1 = \lceil ck/\epsilon \rceil$. Compute, with probability $\geq 1 - \delta/5$, using the algorithm of Lemma 2.2.17, $\tilde{\tau}_i^k(\mathbf{A}^{1/2})$ and $\tilde{\tau}_i^{c'k_1}(\mathbf{A}^{1/2})$ satisfying for all $i \in [n]$:
   - $\tau_i^k(\mathbf{A}^{1/2}) \leq \tilde{\tau}_i^k(\mathbf{A}^{1/2}) \leq 3\tau_i^k(\mathbf{A}^{1/2})$.
   - $\tau_i^{c'k_1}(\mathbf{A}^{1/2}) \leq \tilde{\tau}_i^{c'k_1}(\mathbf{A}^{1/2}) \leq 3\tau_i^{c'k_1}(\mathbf{A}^{1/2})$.

2. Set $\ell_i^{(1)} = \sqrt{\frac{n}{k}}\tilde{\tau}_i^k(\mathbf{A}^{1/2}) + \sqrt{\frac{n\epsilon^4}{k_1}}\tilde{\tau}_i^{c'k_1}(\mathbf{A}^{1/2})$ and $\ell_i^{(2)} = \sqrt{\frac{n}{k_1}}\tilde{\tau}_i^{c'k_1}(\mathbf{A}^{1/2})$. Set $p_i^{(1)} = \frac{\ell_i^{(1)}}{\sum_i \ell_i^{(1)}}$ and $p_i^{(2)} = \frac{\ell_i^{(2)}}{\sum_i \ell_i^{(2)}}$.

3. Set $t_1 = \frac{c_1 \log n}{\delta \cdot \epsilon^2} \sum_i \ell_i^{(1)}$ and $t_2 = \frac{c_2 \log n}{\delta} \sum_i \ell_i^{(2)}$. Sample $\mathbf{S}_1 \in \mathbb{R}^{n \times t_1}$ whose j^{th} column is set to $\frac{1}{\sqrt{tp_i^{(1)}}}\mathbf{e}_i$ with probability $p_i^{(1)}$. Sample $\mathbf{S}_2 \in \mathbb{R}^{n \times t_2}$ analogously using $p_i^{(2)}$.

4. Let $\tilde{\mathbf{A}} = \mathbf{S}_2^T \mathbf{A} \mathbf{S}_1$, and use an input sparsity time algorithm [CW13] to compute orthonormal $\mathbf{Z} \in \mathbb{R}^{t_1 \times k_1}$ satisfying, with probability $\geq 1 - \delta/5$ the spectral guarantee:
$$\|\tilde{\mathbf{A}} - \tilde{\mathbf{A}}\mathbf{Z}\mathbf{Z}^T\|_2^2 \leq 2\|\tilde{\mathbf{A}} - \tilde{\mathbf{A}}_{k_1}\|_2^2 + \frac{\delta}{k_1}\|\tilde{\mathbf{A}} - \tilde{\mathbf{A}}_{k_1}\|_F^2.$$

5. Let $t_3 = c_3\left(\frac{k\log(k/\epsilon)}{\epsilon} + \frac{k}{\delta \cdot \epsilon^2}\right)$, set $p_i^{(3)} = \frac{\|\mathbf{z}_i\|_2^2}{\|\mathbf{Z}\|_F^2}$, and sample $\mathbf{S}_3 \in \mathbb{R}^{t_1 \times t_3}$ where the j^{th} column is set to $\frac{1}{\sqrt{t_3 p_i^{(3)}}}\mathbf{e}_i$ with probability $p_i^{(3)}$. Compute $\mathbf{V} \in \mathbb{R}^{n \times t_3}$ which is an orthonormal basis for the column span of $\mathbf{A}\mathbf{S}_1\mathbf{S}_3$.

6. Let $p_i^{(4)} = \frac{\|\mathbf{v}_i\|_2^2}{\|\mathbf{V}\|_F^2}$ and $t_4 = c_4\left(\frac{t_3 \log t_3}{\epsilon^2} + \frac{t_3}{\delta \cdot \epsilon^2}\right)$. Sample $\mathbf{S}_4 \in \mathbb{R}^{n \times t_4}$ where the j^{th} column is set to $\frac{1}{\sqrt{t_4 p_i^{(4)}}}\mathbf{e}_i$ with probability $p_i^{(4)}$. Compute $\mathbf{W} \in \mathbb{R}^{t_3 \times t_1}$ satisfying:
$$\mathbf{W} = \operatorname*{arg\,min}_{\mathbf{W} \mid \text{rank}(\mathbf{W}) = k} \|\mathbf{S}_4^T \mathbf{A}\mathbf{S}_1\mathbf{S}_3\mathbf{W}\mathbf{Z}^T - \mathbf{S}_4^T\mathbf{A}\mathbf{S}_1\|_F^2.$$

7. Compute an orthonormal basis $\mathbf{M} \in \mathbb{R}^{n \times k}$ for the column span of $\mathbf{A}\mathbf{S}_1\mathbf{S}_3\mathbf{W}$. Let $t_5 = c_5\left(k \log k + \frac{k}{\delta \cdot \epsilon}\right)$, set $p_i^{(5)} = \frac{\|\mathbf{m}_i\|_2^2}{\|\mathbf{M}\|_F^2}$. Sample $\mathbf{S}_5 \in \mathbb{R}^{n \times t_5}$ where the j^{th} column is set of $\frac{1}{\sqrt{t_5 p_i^{(5)}}}\mathbf{e}_i$ with probability $p_i^{(5)}$. Solve:
$$\mathbf{N} = \operatorname*{arg\,min}_{\mathbf{N} \in \mathbb{R}^{n \times k}} \|\mathbf{S}_5^T\mathbf{M}\mathbf{N}^T - \mathbf{S}_5^T\mathbf{A}\|_F^2.$$

8. Return $\mathbf{M}, \mathbf{N} \in \mathbb{R}^{n \times k}$.

The proof of Theorem 2.5.1 breaks down into a number of steps. We first show:

Lemma 2.5.2. *For sufficiently large constants c', c_1, c_2, with probability $\geq 1-\delta/5$,*

1. $\mathbf{AS}_1$ *is an* (ϵ, k)*-PCP of* $\mathbf{A}$ *(Definition 2.1.3).*
2. $\tilde{\mathbf{A}} = \mathbf{S}_2^T\mathbf{AS}_1$ *is a* $(1/2, c_6/\delta, k_1)$*-PCP of* $\mathbf{AS}_1$ *(Definition 2.1.4) for some constant* c_6*.*
3. $\tilde{\mathbf{A}} = \mathbf{S}_2^T\mathbf{AS}_1$ *is a* $(1/2, \alpha, k_1)$*-spectral PCP of* $\mathbf{AS}_1$ *with* $\alpha = \frac{1}{2k_1}\|\mathbf{A}-\mathbf{A}_k\|_F^2$ *(Definition 2.4.2).*

Proof. We prove that each of the three conclusions holds with probability $\geq 1-\delta/15$, giving the lemma via a union bound.

Conclusion 1. In Step 3 of Algorithm 1, $\mathbf{S}_1$ is sampled using probabilities proportional to the scores

$$\ell_i^{(1)} = \sqrt{\frac{n}{k}}\tilde{\tau}_i^k(\mathbf{A}^{1/2}) + \sqrt{\frac{n\epsilon^4}{k_1}}\tilde{\tau}_i^{c'k_1}(\mathbf{A}^{1/2}). \tag{2.53}$$

It contains $t_1 = \frac{c_1 \log n}{\delta \cdot \epsilon^2} \cdot \sum_i \ell_i^{(1)}$ columns.

Applying Lemma 2.3.1, we have that for all i, $\ell_i^{(1)} \geq \frac{\tau_i^k(\mathbf{A})}{2}$. Thus, for sufficiently large c_1, with probability $\geq 1-\delta/15$, $\mathbf{AS}_1$ is an (ϵ,k)-PCP of $\mathbf{A}$.

Conclusion 2. Letting $k' = c'k_1$, we have by (2.53), for all i, $\ell_i^{(1)} \geq \epsilon^2 \cdot \sqrt{\frac{n}{k_1}}\tau_i^{k'}(\mathbf{A}^{1/2})$. If we define $\bar{\ell}_i^{(1)} = \frac{1}{\epsilon^2}\cdot \ell_i^{(1)}$ we have $\bar{\ell}_i^{(1)} = \Omega\left(\sqrt{\frac{n}{k'}}\tau_i^{k'}(\mathbf{A}^{1/2})\right)$ and $t_1 = \frac{c_1 \log n}{\delta}\sum_i \bar{\ell}_i^{(1)}$. Similarly,

$$\ell_i^{(2)}\sqrt{\frac{n}{k_1}}\tilde{\tau}_i^{c'k_1}(\mathbf{A}^{1/2}) = \Omega\left(\sqrt{\frac{n}{k'}}\tau_i^{k'}(\mathbf{A}^{1/2})\right) \tag{2.54}$$

and $t_2 = \frac{c_2 \log n}{\delta}\sum_i \ell_i^{(2)}$. In combination, by Lemma 2.4.1, if c, c_1, c_2 are set large enough, with probability $\geq 1-\delta/15$, $\tilde{\mathbf{A}} = \mathbf{S}_2^T\mathbf{AS}_1$ is a $(1/2, c_6/\delta, k_1)$-PCP of $\mathbf{AS}_1$ for some constant c_6.

Conclusion 3. Again letting $k' = c'k_1$ and $\bar{\ell}_i^{(1)} = \frac{1}{\epsilon^2}\cdot \ell_i^{(1)}$, $\bar{\ell}_i^{(1)} = \Omega\left(\sqrt{\frac{n}{k_1}}\tau_i^{k'}(\mathbf{A}^{1/2})\right)$ and $\ell_i^{(2)} = \Omega\left(\sqrt{\frac{n}{k_1}}\tau_i^{k'}(\mathbf{A}^{1/2})\right)$. We also have $t_1 = \frac{c_1 \log n}{\delta}\sum_i \bar{\ell}_i^{(1)}$ and $t_2 = \frac{c_2 \log n}{\delta}\sum_i \ell_i^{(2)}$, so if c', c_1, c_2 are set large enough, by Lemma 2.4.3, with probability $\geq 1-\delta/15$, $\tilde{\mathbf{A}}$ is a $(1/2, \alpha, k_1)$-spectral PCP of $\mathbf{AS}_1$ with $\alpha = \frac{1}{2k_1}\|\mathbf{A}-\mathbf{A}_k\|_F^2$. □

We can use the PCP properties shown in Lemma 2.5.2 to show:

Lemma 2.5.3. *For sufficiently large constants c, c', c_1, c_2, with probability $\geq 1 - 2\delta/5$, all bounds of Lemma 2.5.2 hold and further, Steps 1-4 of Algorithm 1 produce orthonormal $\mathbf{Z} \in \mathbb{R}^{t_1 \times k_1}$ satisfying:*

$$\|\mathbf{AS}_1 - \mathbf{AS}_1\mathbf{ZZ}^T\|_2^2 \leq \frac{\epsilon}{k}\|\mathbf{A} - \mathbf{A}_k\|_F^2.$$

Proof. With probability $\geq 1 - \delta/5$, all three conclusions of Lemma 2.5.2 hold. Let $\mathbf{V}_{k_1}$ contain the top k_1 row singular vectors of $\mathbf{AS}_1$ so $(\mathbf{AS}_1)_{k_1} = \mathbf{AS}_1\mathbf{V}_{k_1}\mathbf{V}_{k_1}^T$. By conclusion (2), for some constant c_6:

$$\begin{aligned}\|\tilde{\mathbf{A}} - \tilde{\mathbf{A}}_{k_1}\|_F^2 &\leq \|\tilde{\mathbf{A}} - \tilde{\mathbf{A}}\mathbf{V}_{k_1}\mathbf{V}_{k_1}^T\|_F^2\\ &\leq \frac{3}{2}\|\mathbf{AS}_1 - (\mathbf{AS}_1)_{k_1}\|_F^2 - E\\ &\leq (3/2 + c_6/\delta)\|\mathbf{AS}_1 - (\mathbf{AS}_1)_{k_1}\|_F^2. \end{aligned} \tag{2.55}$$

By conclusion (1), $\mathbf{AS}_1$ is an (ϵ, k)-PCP of $\mathbf{A}$, which gives:

Claim 2.5.4. *If $\mathbf{AS}_1$ is an (ϵ, k)-PCP of $\mathbf{A}$ then:*

$$\|\mathbf{AS}_1 - (\mathbf{AS}_1)_k\|_F^2 \leq (1+\epsilon)\|\mathbf{A} - \mathbf{A}_k\|_F^2.$$

Proof. Letting $\mathbf{Z}$ contain as its columns $\mathbf{A}$'s top k left singular vectors,

$$\begin{aligned}\|\mathbf{AS}_1 - (\mathbf{AS}_1)_k\|_F^2 &\leq \|\mathbf{AS}_1 - \mathbf{ZZ}^T\mathbf{AS}_1\|_F^2\\ &\leq (1+\epsilon)\|\mathbf{A} - \mathbf{ZZ}^T\mathbf{A}\|_F^2\\ &= (1+\epsilon)\|\mathbf{A} - \mathbf{A}_k\|_F^2,\end{aligned}$$

which gives the claim. □

Combining Claim 2.5.4 with (2.55) we have:

$$\begin{aligned}\|\tilde{\mathbf{A}} - \tilde{\mathbf{A}}_{k_1}\|_F^2 &\leq (3/2 + c_6/\delta)\|\mathbf{AS}_1 - (\mathbf{AS}_1)_{k_1}\|_F^2\\ &\leq (3/2 + c_6/\delta)\|\mathbf{AS}_1 - (\mathbf{AS}_1)_k\|_F^2\\ &\leq (3/2 + c_6/\delta)(1+\epsilon)\|\mathbf{A} - \mathbf{A}_k\|_F^2.\end{aligned} \tag{2.56}$$

Further, via conclusion (3) of Lemma 2.5.2, for any rank-k_1 orthogonal projection matrix $\mathbf{P} \in \mathbb{R}^{n\times n}$:

$$\frac{1}{2}\|\mathbf{A}\mathbf{S}_1(\mathbf{I}-\mathbf{P})\|_2^2 - \frac{1}{2k_1}\|\mathbf{A}-\mathbf{A}_{k_1}\|_F^2 \leq \|\tilde{\mathbf{A}}(\mathbf{I}-\mathbf{P})\|_2^2 \leq \frac{3}{2}\|\mathbf{A}\mathbf{S}_1(\mathbf{I}-\mathbf{P})\|_2^2 + \frac{1}{2k_1}\|\mathbf{A}-\mathbf{A}_{k_1}\|_F^2. \tag{2.57}$$

By (2.57), letting $\mathbf{V}_{k_1}$ contain the top k_1 row singular vectors of $\mathbf{A}\mathbf{S}_1$:

$$\begin{aligned}\|\tilde{\mathbf{A}} - \tilde{\mathbf{A}}_{k_1}\|_2^2 &\leq \|\tilde{\mathbf{A}} - \tilde{\mathbf{A}}\mathbf{V}_{k_1}\mathbf{V}_{k_1}^T\|_2^2\\ &\leq \frac{3}{2}\|\mathbf{A}\mathbf{S}_1 - (\mathbf{A}\mathbf{S}_1)_{k_1}\|_2^2 + \frac{1}{2k_1}\|\mathbf{A}-\mathbf{A}_{k_1}\|_F^2\\ &\leq \frac{3\epsilon}{k}\|\mathbf{A}-\mathbf{A}_k\|_F^2, \end{aligned} \tag{2.58}$$

where the final bound follows because if we set $c \geq 2$, $k_1 \geq \lceil 2k/\epsilon \rceil$ so

$$\|\mathbf{A}\mathbf{S}_1 - (\mathbf{A}\mathbf{S}_1)_{k_1}\|_2^2 \leq \frac{\epsilon}{k}\|\mathbf{A}\mathbf{S}_1 - (\mathbf{A}\mathbf{S}_1)_k\|_F^2 \leq \frac{(1+\epsilon)\cdot\epsilon}{k}\|\mathbf{A}-\mathbf{A}_k\|_F^2$$

by Claim 2.5.4. For $\epsilon \in (0, 1/2]$ this gives:

$$\frac{3}{2}\|\mathbf{A}\mathbf{S}_1-(\mathbf{A}\mathbf{S}_1)_{k_1}\|_2^2+\frac{1}{2k_1}\|\mathbf{A}-\mathbf{A}_{k_1}\|_F^2 \leq \left(\frac{3(1+\epsilon)}{2}+\frac{1}{2}\right)\cdot\frac{\epsilon}{k}\|\mathbf{A}-\mathbf{A}_k\|_F^2 \leq \frac{3\epsilon}{k}\|\mathbf{A}-\mathbf{A}_k\|_F^2.$$

With (2.56) and (2.58) in place, applying (2.57) again, if we compute $\mathbf{Z} \in \mathbb{R}^{t_1\times k_1}$ satisfying the guarantee of Step 4 with probability $\geq 1-\delta/5$, then via a union bound, with probability $\geq 1-2\delta/5$:

$$\begin{aligned}\|\mathbf{A}\mathbf{S}_1 - \mathbf{A}\mathbf{S}_1\mathbf{Z}\mathbf{Z}^T\|_2^2 &\leq 2\|\tilde{\mathbf{A}} - \tilde{\mathbf{A}}\mathbf{Z}\mathbf{Z}^T\|_2^2 + \frac{1}{k_1}\|\mathbf{A}-\mathbf{A}_{k_1}\|_F^2\\ &\leq 4\|\tilde{\mathbf{A}}-\tilde{\mathbf{A}}_{k_1}\|_2^2 + \frac{2}{\delta\cdot k_1}\|\tilde{\mathbf{A}}-\tilde{\mathbf{A}}_{k_1}\|_F^2 + \frac{1}{k_1}\|\mathbf{A}-\mathbf{A}_{k_1}\|_F^2\\ &\qquad\text{(By the guarantee on } \mathbf{Z} \text{ in Step 4)}\\ &\leq (3+\delta\cdot(3/2+c_6/\delta)(1+\epsilon)+1)\cdot\frac{\epsilon}{k}\cdot\|\mathbf{A}-\mathbf{A}_k\|_F^2\\ &\leq (2c_6+7)\cdot\frac{\epsilon}{k}\|\mathbf{A}-\mathbf{A}_k\|_F^2.\end{aligned}$$

The lemma follows after adjusting constants on ϵ by making c, c', c_1 and c_2 sufficiently large. □

Lemma 2.5.3 ensures that the rank-k matrix $\mathbf{W}$ computed in Step 6 gives a near-optimal low-rank approximation of $\mathbf{A}\mathbf{S}_1$. Specifically:

Lemma 2.5.5. *With probability $\geq 1-4\delta/5$, all bounds of Lemma 2.5.2 and 2.5.3 hold and further, for sufficiently large $c, c', c_1, c_2, c_3, c_4$, Step 5 of Algorithm 1 produces $\mathbf{W}$ such that, letting $\mathbf{M} \in \mathbb{R}^{n \times k}$ be an orthonormal basis for the column span of $\mathbf{AS}_1\mathbf{S}_3\mathbf{W}$:*

$$\|\mathbf{AS}_1 - \mathbf{MM}^T\mathbf{AS}_1\|_F^2 \leq (1+\epsilon)\|\mathbf{A} - \mathbf{A}_k\|_F^2.$$

Proof. We first consider the optimization problem:

$$\mathbf{T}^* = \operatorname*{arg\,min}_{\mathbf{T} |\operatorname{rank}(\mathbf{T})=k} \|\mathbf{TZ}^T - \mathbf{AS}_1\|_F^2.$$

We have, by Lemma 2.5.3, with probability $\geq 1 - 2\delta/5$:

$$\begin{aligned}
\|\mathbf{T}^*\mathbf{Z}^T - \mathbf{AS}_1\|_F^2 &\leq \|(\mathbf{AS}_1)_k\mathbf{ZZ}^T - \mathbf{AS}_1\|_F^2 \\
&= \|[\mathbf{AS}_1 - (\mathbf{AS}_1)_k] + (\mathbf{AS}_1)_k(\mathbf{I} - \mathbf{ZZ}^T)\|_F^2 \\
&= \|\mathbf{AS}_1 - (\mathbf{AS}_1)_k\|_F^2 + \|(\mathbf{AS}_1)_k(\mathbf{I} - \mathbf{ZZ}^T)\|_F^2 \\
&\leq \|\mathbf{AS}_1 - (\mathbf{AS}_1)_k\|_F^2 + k \cdot \|\mathbf{AS}_1(\mathbf{I} - \mathbf{ZZ}^T)\|_2^2 \\
&\leq \|\mathbf{AS}_1 - (\mathbf{AS}_1)_k\|_F^2 + \epsilon\|\mathbf{A} - \mathbf{A}_k\|_F^2 \\
&\leq (1+2\epsilon)\|\mathbf{A} - \mathbf{A}_k\|_F^2, \qquad (2.59)
\end{aligned}$$

where the second to last step uses that $\|\mathbf{AS}_1(\mathbf{I} - \mathbf{ZZ}^T)\|_2^2 \leq \frac{\epsilon}{k}\|\mathbf{A} - \mathbf{A}_k\|_F^2$ by Lemma 2.5.3 and the last step uses that $\|\mathbf{AS}_1 - (\mathbf{AS}_1)_k\|_F^2 \leq (1+\epsilon)\|\mathbf{A} - \mathbf{A}_k\|_F^2$, which follows from conclusion (1) of Lemma 2.5.2 ($\mathbf{AS}_1$ is an (ϵ, k)-PCP of $\mathbf{A}$) and Claim 2.5.4.

Note that since it is rank-k we can write $\mathbf{T}^* = \mathbf{Y}^*\mathbf{N}^*$ where $\mathbf{Y}^* \in \mathbb{R}^{n \times k}$ and $\mathbf{N}^* \in \mathbb{R}^{k \times k_1}$ has orthonormal rows (spanning the rows of $\mathbf{T}^*$). $\mathbf{Y}^*$ is the solution to the unconstrained low-rank approximation problem:

$$\mathbf{Y}^* = \operatorname*{arg\,min}_{\mathbf{Y} \in \mathbb{R}^{n \times k}} \|\mathbf{YN}^*\mathbf{Z}^T - \mathbf{AS}_1\|_F^2.$$

Since both $\mathbf{N}^*$ and $\mathbf{Z}^T$ (for $\mathbf{Z}$ computed in Step 4) have orthonormal rows, so does $\mathbf{N}^*\mathbf{Z}^T$. so its column norms are its leverage scores. $\mathbf{S}_3$ is sampled using the column norms of $\mathbf{Z}^T$, which upper bound those of $\mathbf{N}^*\mathbf{Z}^T$. Since $\|\mathbf{Z}\|_F^2 = k_1 = \lceil ck/\epsilon \rceil$ and since $\operatorname{rank}(\mathbf{N}^*\mathbf{Z}^T) \leq \operatorname{rank}(\mathbf{Z}) = k_1 = \lceil ck/\epsilon \rceil$, we have

$$t_3 = \Omega\left(\|\mathbf{Z}\|_F^2 \cdot \log(\operatorname{rank}(\mathbf{N}^*\mathbf{Z}^T)) + \frac{\|\mathbf{Z}\|_F^2}{\delta \cdot \epsilon}\right).$$

Thus, by the approximate regression result of Lemma 2.2.14 if we set:

$$\tilde{\mathbf{Y}} = \operatorname*{arg\,min}_{\mathbf{Y} \in \mathbb{R}^{n \times k}} \|\mathbf{Y}\mathbf{N}^*\mathbf{Z}^T\mathbf{S}_3 - \mathbf{A}\mathbf{S}_1\mathbf{S}_3\|_F^2$$

if the constant c_3 is large enough, with probability $\geq 1 - \delta/5$:

$$\|\tilde{\mathbf{Y}}\mathbf{N}^*\mathbf{Z}^T - \mathbf{A}\mathbf{S}_1\|_F^2 \leq (1+\epsilon)\|\mathbf{Y}^*\mathbf{N}^*\mathbf{Z}^T - \mathbf{A}\mathbf{S}_1\|_F^2. \tag{2.60}$$

Given $\mathbf{N}^*$, $\tilde{\mathbf{Y}}$ can be computed in closed form as $\tilde{\mathbf{Y}} = \mathbf{A}\mathbf{S}_1\mathbf{S}_3(\mathbf{N}^*\mathbf{Z}^T\mathbf{S}_3)^+$, which is in the column span of $\mathbf{A}\mathbf{S}_1\mathbf{S}_3$. Thus (2.60) demonstrates that, with probability $\geq 1 - 3\delta/5$ there is some rank-k $\mathbf{T}$ in this span satisfying

$$\|\mathbf{T}\mathbf{Z}^T - \mathbf{A}\mathbf{S}_1\|_F^2 \leq (1+\epsilon)\|\mathbf{T}^*\mathbf{Z}^T - \mathbf{A}\mathbf{S}_1\|_F^2 \leq (1+4\epsilon)\|\mathbf{A} - \mathbf{A}_k\|_F^2$$

by (2.59) and a union bound over the probability that (2.59) and (2.60) hold. Thus if we compute

$$\mathbf{W}^* = \operatorname*{arg\,min}_{\mathbf{W} \mid \operatorname{rank}(\mathbf{W})=k} \|\mathbf{A}\mathbf{S}_1\mathbf{S}_3\mathbf{W}\mathbf{Z}^T - \mathbf{A}\mathbf{S}_1\|_F^2$$

then we have with probability $\geq 1 - 3\delta/5$:

$$\|\mathbf{A}\mathbf{S}_1\mathbf{S}_3\mathbf{W}^*\mathbf{Z}^T - \mathbf{A}\mathbf{S}_1\|_F^2 \leq (1+4\epsilon)\|\mathbf{A} - \mathbf{A}_k\|_F^2. \tag{2.61}$$

We compute $\mathbf{W}^*$ approximately in Step 6 by sampling the rows of $\mathbf{A}\mathbf{S}_1\mathbf{S}_3$ by their leverage scores. By Lemma 2.2.5, these are given by the row norms of the orthonormal basis $\mathbf{V}$ as computed in Step 5 and sampled with to obtain $\mathbf{S}_4$ in Step 6. We claim:

Claim 2.5.6. *For*

$$\tilde{\mathbf{W}} = \operatorname*{arg\,min}_{\mathbf{W} \mid \operatorname{rank}(\mathbf{W})=k} \|\mathbf{S}_4^T\mathbf{A}\mathbf{S}_1\mathbf{S}_3\mathbf{W}\mathbf{Z}^T - \mathbf{S}_4^T\mathbf{A}\mathbf{S}_1\|_F^2,$$

with probability $\geq 1 - \delta/5$*,*

$$\|\mathbf{A}\mathbf{S}_1\mathbf{S}_3\mathbf{W}\mathbf{Z}^T - \mathbf{A}\mathbf{S}_1\|_F^2 \leq (1+\epsilon)\|\mathbf{A}\mathbf{S}_1\mathbf{S}_3\mathbf{W}^*\mathbf{Z}^T - \mathbf{A}\mathbf{S}_1\|_F^2.$$

Proof. For any $\mathbf{W}$, by the Pythagorean theorem, since $\mathbf{V}$ spans $\mathbf{A}\mathbf{S}_1\mathbf{S}_3$,

$$\|\mathbf{A}\mathbf{S}_1\mathbf{S}_3\mathbf{W}\mathbf{Z}^T - \mathbf{A}\mathbf{S}_1\|_F^2 = \|\mathbf{A}\mathbf{S}_1\mathbf{S}_3\mathbf{W}\mathbf{Z}^T - \mathbf{V}\mathbf{V}^T\mathbf{A}\mathbf{S}_1\|_F^2 + \|(\mathbf{I} - \mathbf{V}\mathbf{V}^T)\mathbf{A}\mathbf{S}_1\|_F^2. \tag{2.62}$$

Similarly

$$\begin{aligned}\|\mathbf{S}_4^T\mathbf{A}\mathbf{S}_1\mathbf{S}_3\mathbf{W}\mathbf{Z}^T - \mathbf{S}_4^T\mathbf{A}\mathbf{S}_1\|_F^2 = \|\mathbf{S}_4^T\mathbf{A}\mathbf{S}_1\mathbf{S}_3\mathbf{W}\mathbf{Z}^T - \mathbf{S}_4^T\mathbf{V}\mathbf{V}^T\mathbf{A}\mathbf{S}_1\|_F^2 + \|\mathbf{S}_4^T(\mathbf{I}-\mathbf{V}\mathbf{V}^T)\mathbf{A}\mathbf{S}_1\|_F^2\\ + 2\,\mathrm{tr}\left((\mathbf{Z}\mathbf{W}^T\mathbf{S}_3^T\mathbf{S}_1^T\mathbf{A}^T - \mathbf{S}_1^T\mathbf{A}^T\mathbf{V}\mathbf{V}^T)\mathbf{S}_4\mathbf{S}_4^T(\mathbf{I}-\mathbf{V}\mathbf{V}^T)\mathbf{A}\mathbf{S}_1\right).\end{aligned} \tag{2.63}$$

For the first term, since $\mathbf{S}_4$ is sampled via the leverage scores of $\mathbf{A}\mathbf{S}_1\mathbf{S}_3$ and $t_4 = c_4\left(\frac{t_3\log t_3}{\epsilon^2} + \frac{t_3}{\delta\cdot\epsilon^2}\right)$, by Lemma 2.2.6, if c_4 is sufficiently large, $\mathbf{S}_4$ gives a subspace embedding for the column span of $\mathbf{A}\mathbf{S}_1\mathbf{S}_3$ with probability $\geq 1-\delta/10$. That is,

$$\|\mathbf{S}_4^T\mathbf{A}\mathbf{S}_1\mathbf{S}_3\mathbf{W}\mathbf{Z}^T - \mathbf{S}_4^T\mathbf{V}\mathbf{V}^T\mathbf{A}\mathbf{S}_1\|_F^2 \in (1\pm\epsilon)\|\mathbf{A}\mathbf{S}_1\mathbf{S}_3\mathbf{W}\mathbf{Z}^T - \mathbf{V}\mathbf{V}^T\mathbf{A}\mathbf{S}_1\|_F^2. \tag{2.64}$$

For the cross term, again since $\mathbf{V}$ spans the columns of $\mathbf{A}\mathbf{S}_1\mathbf{S}_3$ we have:

$$\begin{aligned}&\mathrm{tr}\left((\mathbf{Z}\mathbf{W}^T\mathbf{S}_3^T\mathbf{S}_1^T\mathbf{A}^T - \mathbf{S}_1^T\mathbf{A}^T\mathbf{V}\mathbf{V}^T)\mathbf{S}_4\mathbf{S}_4^T(\mathbf{I}-\mathbf{V}\mathbf{V}^T)\mathbf{A}\mathbf{S}_1\right)\\ &\qquad= \mathrm{tr}\left((\mathbf{Z}\mathbf{W}^T\mathbf{S}_3^T\mathbf{S}_1^T\mathbf{A}^T - \mathbf{S}_1^T\mathbf{A}^T)\mathbf{V}\mathbf{V}^T\mathbf{S}_4\mathbf{S}_4^T(\mathbf{I}-\mathbf{V}\mathbf{V}^T)\mathbf{A}\mathbf{S}_1\right)\\ &\qquad\leq \|\mathbf{A}\mathbf{S}_1\mathbf{S}_3\mathbf{W}\mathbf{Z}^T - \mathbf{A}\mathbf{S}_1\|_F\cdot\|\mathbf{V}\mathbf{V}^T\mathbf{S}_4\mathbf{S}_4^T(\mathbf{I}-\mathbf{V}\mathbf{V}^T)\mathbf{A}\mathbf{S}_1\|_F\\ &\qquad\leq \|\mathbf{A}\mathbf{S}_1\mathbf{S}_3\mathbf{W}\mathbf{Z}^T - \mathbf{A}\mathbf{S}_1\|_F\cdot\epsilon\|(\mathbf{I}-\mathbf{V}\mathbf{V}^T)\mathbf{A}\mathbf{S}_1\|_F,\end{aligned}$$

where the last bound holds with probability $\geq 1-\delta/10$ if c_4 is set large enough by the approximate matrix multiplcaiton result of Lemma 2.2.16. We can bound

$$\|(\mathbf{I}-\mathbf{V}\mathbf{V}^T)\mathbf{A}\mathbf{S}_1\|_F \leq \|\mathbf{A}\mathbf{S}_1\mathbf{S}_3\mathbf{W}^*\mathbf{Z}^T - \mathbf{A}\mathbf{S}_1\|_F \leq \|\mathbf{A}\mathbf{S}_1\mathbf{S}_3\mathbf{W}\mathbf{Z}^T - \mathbf{A}\mathbf{S}_1\|_F$$

and so overall, with probability $\geq 1-\delta/10$:

$$\mathrm{tr}\left((\mathbf{Z}\mathbf{W}^T\mathbf{S}_3^T\mathbf{S}_1^T\mathbf{A}^T - \mathbf{S}_1^T\mathbf{A}^T\mathbf{V}\mathbf{V}^T)\mathbf{S}_4\mathbf{S}_4^T(\mathbf{I}-\mathbf{V}\mathbf{V}^T)\mathbf{A}\mathbf{S}_1\right) \leq \epsilon\|\mathbf{A}\mathbf{S}_1\mathbf{S}_3\mathbf{W}\mathbf{Z}^T - \mathbf{A}\mathbf{S}_1\|_F^2. \tag{2.65}$$

Combining (2.62), (2.63), (2.64), and (2.65), with probability $\geq 1-\delta/5$ for all $\mathbf{W}$,

$$\|\mathbf{S}_4^T\mathbf{A}\mathbf{S}_1\mathbf{S}_3\mathbf{W}\mathbf{Z}^T - \mathbf{S}_4^T\mathbf{A}\mathbf{S}_1\|_F^2 \in (1\pm 3\epsilon)\|\mathbf{A}\mathbf{S}_1\mathbf{S}_3\mathbf{W}\mathbf{Z}^T - \mathbf{A}\mathbf{S}_1\|_F^2 + \Delta,$$

where $\Delta = \|\mathbf{S}_4^T(\mathbf{I}-\mathbf{V}\mathbf{V}^T)\mathbf{A}\mathbf{S}_1\|_F^2 - \|(\mathbf{I}-\mathbf{V}\mathbf{V}^T)\mathbf{A}\mathbf{S}_1\|_F^2$ is fixed independent of $\mathbf{W}$.

Thus, since $\tilde{\mathbf{W}} = \|\mathbf{S}_4^T\mathbf{A}\mathbf{S}_1\mathbf{S}_3\mathbf{W}\mathbf{Z}^T - \mathbf{S}_4^T\mathbf{A}\mathbf{S}_1\|_F^2$, with probability $\geq 1-\delta/5$,

$$\|\mathbf{A}\mathbf{S}_1\mathbf{S}_3\tilde{\mathbf{W}}\mathbf{Z}^T - \mathbf{A}\mathbf{S}_1\|_F^2 \leq \frac{(1+3\epsilon)}{(1-3\epsilon)}\|\mathbf{A}\mathbf{S}_1\mathbf{S}_3\mathbf{W}^*\mathbf{Z}^T - \mathbf{A}\mathbf{S}_1\|_F^2,$$

which gives the claim after adjusting constants on ϵ by making c_4 sufficiently large. □

Combining Claim 2.5.6 with (2.61) gives that, with probability $\geq 1 - 4\delta/5$:

$$\|\mathbf{A}\mathbf{S}_1\mathbf{S}_3\tilde{\mathbf{W}}\mathbf{Z}^T - \mathbf{A}\mathbf{S}_1\|_F^2 \leq (1+\epsilon)(1+4\epsilon)\|\mathbf{A} - \mathbf{A}_k\|_F^2.$$

Further, since $\mathbf{M} \in \mathbb{R}^{n\times k}$ is an orthonormal span of $\mathbf{A}\mathbf{S}_1\mathbf{S}_3\mathbf{W}$,

$$\|\mathbf{A}\mathbf{S}_1 - \mathbf{M}\mathbf{M}^T\mathbf{A}\mathbf{S}_1\|_F^2 \leq \|\mathbf{A}\mathbf{S}_1 - \mathbf{A}\mathbf{S}_1\mathbf{S}_3\tilde{\mathbf{W}}\mathbf{Z}^T\|_F^2 \leq (1+\epsilon)(1+4\epsilon)\|\mathbf{A} - \mathbf{A}_k\|_F^2.$$

This gives the lemma after adjusting constants on ϵ by making $c, c', c_1, c_2, c_3, c_4$ large enough. □

We finally can prove our main theorem, showing the correctness and runtime of Algorithm 1.

Proof of Theorem 2.5.1. We first show the correctness of the algorithm with high probability. We then discuss its runtime.

Correctness

With probability $\geq 1-4\delta/5$ all bounds of Lemmas 2.5.2, 2.5.3 and 2.5.5 hold. By the bound of Lemma 2.5.5, and conclusion (1) of Lemma 2.5.2 (that $\mathbf{A}\mathbf{S}_1$ is an (ϵ, k)-PCP of $\mathbf{A}$),

$$\|\mathbf{A} - \mathbf{M}\mathbf{M}^T\mathbf{A}\|_F^2 \leq \frac{1+\epsilon}{1-\epsilon}\|\mathbf{A} - \mathbf{A}_k\|_F^2.$$

In Step 7 we sample $c_5\left(k\log k + \frac{k}{\delta\cdot\epsilon}\right)$ rows of $\mathbf{M}$ by their standard leverage scores (their row norms) and so by the approximate regression result of Lemma 2.2.14, if c_5 is set large enough, have with probability $\geq 1 - \delta/5$:

$$\begin{aligned}\|\mathbf{M}\mathbf{N}^T - \mathbf{A}\|_F^2 &\leq (1+\epsilon)\min_{\mathbf{Y}\in\mathbb{R}^{n\times k}}\|\mathbf{M}\mathbf{Y}^T - \mathbf{A}\|_F^2\\ &= (1+\epsilon)\|\mathbf{A} - \mathbf{M}\mathbf{M}^T\mathbf{A}\|_F^2\\ &\leq \frac{(1+\epsilon)^2}{1-\epsilon}\|\mathbf{A} - \mathbf{A}_k\|_F^2.\end{aligned}$$

After adjusting constants on ϵ by setting $c, c', c_1, c_2, c_3, c_4, c_5$ large enough, and union bounding, with probability $\geq 1-\delta$, $\|\mathbf{M}\mathbf{N}^T - \mathbf{A}\|_F^2 \leq (1+\epsilon)\|\mathbf{A} - \mathbf{A}_k\|_F^2$, which gives the correctness of the algorithm.

Runtime and Query Complexity

It just remains to discuss Algorithm 1's runtime and query complexity. We begin with the query complexity:

- In Step 1 computing the leverage score approximations requires $O(nk_1 \log(k_1/\delta)) = O\left(\frac{nk\log(k/(\delta\epsilon))}{\epsilon}\right)$ accesses to $\mathbf{A}$ by Lemma 2.2.17.

- In Step 4 forming $\tilde{\mathbf{A}} = \mathbf{S}_2^T\mathbf{A}\mathbf{S}_1$ requires accessing $t_1 \cdot t_2 = O\left(\frac{nk\log^2 n}{\delta^2 \cdot \epsilon^{2.5}}\right)$ entries of $\mathbf{A}$.

- In Step 5, forming $\mathbf{A}\mathbf{S}_1\mathbf{S}_3$ requires accessing $n \cdot t_3 = O\left(\frac{nk\log(k/\epsilon)}{\epsilon} + \frac{nk}{\delta \cdot \epsilon^2}\right)$ entries of $\mathbf{A}$.

- In Step 6, forming $\mathbf{S}_4^T\mathbf{A}\mathbf{S}_1$ requires $t_1 \cdot t_4$ accesses to $\mathbf{A}$. We have

$$t_4 = O\left(\frac{t_3 \log t_3}{\epsilon^2} + \frac{t_3}{\delta \cdot \epsilon^2}\right) = O\left(\frac{k\log(k/\epsilon)}{\delta^2\epsilon^2}\right)$$

  and so $t_1 \cdot t_4 = O\left(\frac{\sqrt{n}k^{1.5}\log(k/\epsilon)\log n}{\delta^3 \cdot \epsilon^6}\right)$. When n is large compared to k, $1/\epsilon$, and $1/\delta$ this term will be dominated by our linear in n terms.

- In Step 7, forming $\mathbf{S}_5^T\mathbf{A}$ requires $n \cdot t_5 = O\left(nk\log k + \frac{nk}{\delta \cdot \epsilon}\right)$ accesses to $\mathbf{A}$.

Overall the access complexity is dominated by the accesses in Step 4 and potentially Step 6, giving us total access complexity

$$O\left(\frac{n \cdot k\log^2(n)}{\delta^2 \cdot \epsilon^{2.5}} + \sqrt{n}k^{1.5} \cdot \log n \cdot \log k \cdot \text{poly}(1/\epsilon, 1/\delta)\right).$$

We next consider time complexity:

- In Step 1, computing the leverage score approximations requires

$$O(n(k_1\log(k_1/\delta))^{\bar{\omega}-1}) = \tilde{O}\left(\frac{nk^{\bar{\omega}-1}}{\epsilon^{\bar{\omega}-1}}\right)$$

  time by Lemma 2.2.17.

- In Step 4, computing $\mathbf{Z}$ can be done using an input sparsity time algorithm for spectral norm error with rank k_1 and error parameter $\epsilon' = \Theta(\delta)$. By Theorem 27 of [CEM+15] or using input sparsity time ridge leverage score sampling

[CMM17] in conjunction with the spectral norm PCP result of Lemma 2.6.2, the total runtime required is:

$$O(\mathrm{nnz}(\tilde{\mathbf{A}}))+\tilde{O}\left(\sqrt{n}k^{\bar{\omega}-1}\cdot \mathrm{poly}(1/\epsilon,1/\delta)\right)=O\left(\frac{nk\log^2(n)}{\delta^2\cdot\epsilon^{2.5}}+\sqrt{n}k^{\bar{\omega}-1}\cdot\mathrm{poly}(1/\epsilon,1/\delta)\right).$$

- In Step 5, $\mathbf{V}$ can be computed in $O(nt_3^{\bar{\omega}-1})=\tilde{O}\left(\frac{nk^{\bar{\omega}-1}}{\delta^{\bar{\omega}-1}\cdot\epsilon^{2(\bar{\omega}-1)}}\right)$ time.
- In Step 6, we can compute $\mathbf{W}$ by first multiplying $\mathbf{S}_4^T\mathbf{A}\mathbf{S}_1$ by $\mathbf{Z}$ and then multiplying by $(\mathbf{S}_4^T\mathbf{A}\mathbf{S}_1\mathbf{S}_3)^+$ and taking the best rank-k approximation of the result. The total runtime is

$$\tilde{O}\left(t_1\cdot k^{\bar{\omega}-1}+k^{\bar{\omega}}\right)\cdot\mathrm{poly}(1/\epsilon,1/\delta)=\tilde{O}(\sqrt{n}k^{\bar{\omega}-.5})\cdot\mathrm{poly}(1/\epsilon,1/\delta).$$

- In Step 7 we can compute $\mathbf{M}$ by first computing a $t_3\times k$ span of the column space of $\mathbf{W}$, multiplying this by $\mathbf{A}\mathbf{S}_1\mathbf{S}_3$, and then taking an orthonormal basis for the result. This requires total time

$$\tilde{O}(k^{\bar{\omega}}\cdot\mathrm{poly}(1/\epsilon,1/\delta))+O\left(nk^{\bar{\omega}-1}\left(\frac{\log(k/\epsilon)}{\epsilon}+\frac{1}{\delta\cdot\epsilon^2}\right)+nk^{\bar{\omega}-1}\right).$$

  The final regression problem can be solved by forming the pseudoinverse of $\mathbf{S}_5^T\mathbf{M}$ in $O\left(k^{\bar{\omega}}\left(\log k+\frac{1}{\delta\cdot\epsilon}\right)\right)$ time and applying it to $\mathbf{S}_5^T\mathbf{A}$ in $O\left(nk^{\bar{\omega}-1}\left(\log k+\frac{1}{\delta\cdot\epsilon}\right)\right)$ time.

Overall our runtime cost with linear dependence on n is dominated by the costs of Steps 4 and 5, which we can upper bound by

$$\tilde{O}\left(\frac{nk^{\bar{\omega}-1}}{\delta^2\epsilon^{2(\bar{\omega}-1)}}\right).$$

We additionally must add the $\tilde{O}(\sqrt{n}k^{\bar{\omega}-.5})\cdot\mathrm{poly}(1/\epsilon,1/\delta)$ term from Step 6, which only dominates if n is relatively small. This completes the proof of the theorem. □

2.5.2 Outputting a PSD Matrix

In many applications it is desirable that the low-rank approximation to $\mathbf{A}$ is also symmetric and positive semidefinite. This is equivalent to restricting $\mathbf{M}=\mathbf{N}$ in Problem 2.1.1.

It is not hard to see that a modification to Algorithm 1 can satisfy this constraint also in $\tilde{O}(n\,\mathrm{poly}(k/\epsilon))$ time. The upshot is:

Theorem 2.5.7 (Sublinear Time Low-Rank Approximation – PSD Output)**.** *There is an algorithm that given any PSD $\mathbf{A} \in \mathbb{R}^{n\times n}$, $k \in \mathbb{Z}^{\geq 1}$, and $\epsilon, \delta \in (0, 1/2]$ accesses $\tilde{O}\left(\frac{nk^2}{\delta\cdot\epsilon^2} + \frac{nk}{\delta^2\epsilon^3}\right)$ entries of $\mathbf{A}$, runs in*

$$\tilde{O}\left(\frac{nk^{\bar{\omega}}}{\delta\epsilon^{\bar{\omega}}} + \frac{nk^{\bar{\omega}-1}}{(\delta^2\cdot\epsilon^3)^{(\bar{\omega}-1)}} + (\sqrt{n}k^{\bar{\omega}-1} + k^{\bar{\omega}+1})\cdot\mathrm{poly}(1/\epsilon, 1/\delta)\right)$$

time, and with probability $\geq 1-\delta$ outputs $\mathbf{M} \in \mathbb{R}^{n\times k}$ with:

$$\|\mathbf{A} - \mathbf{M}\mathbf{M}^T\|_F^2 \leq (1+\epsilon)\|\mathbf{A} - \mathbf{A}_k\|_F^2.$$

We start with the following lemma, which shows that, if we have a good low-rank subspace for approximating $\mathbf{A}$ in the spectral norm (computable using Theorem 2.6.3), then we can quickly find a near-optimal PSD low-rank approximation:

Lemma 2.5.8. *Given PSD $\mathbf{A} \in \mathbb{R}^{n\times n}$, $k \in \mathbb{Z}^{\geq 1}$, $\epsilon, \delta \in (0, 1/2]$, and orthonormal $\mathbf{Z} \in \mathbb{R}^{n\times m}$ with $\|\mathbf{A} - \mathbf{A}\mathbf{Z}\mathbf{Z}^T\|_2^2 \leq \frac{\epsilon}{k}\|\mathbf{A} - \mathbf{A}_k\|_F^2$, there is an algorithm that accesses $O\left(\frac{nm\log m/\delta}{\delta^2\epsilon^2}\right)$ entries of $\mathbf{A}$, runs in $\tilde{O}\left(\frac{nm^{\bar{\omega}-1}}{(\delta\epsilon)^{2(\bar{\omega}-1)}}\right)$ time, and outputs $\mathbf{M} \in \mathbb{R}^{n\times k}$ that with probability $\geq 1-\delta$ satisfies, for some fixed constant c:*

$$\|\mathbf{A} - \mathbf{M}\mathbf{M}^T\|_F^2 \leq (1+c\epsilon)\|\mathbf{A} - \mathbf{A}_k\|_F^2.$$

Proof. By Lemma 10 of [CW17a], there exists some fixed constant c such that, for any orthonormal $\mathbf{Z} \in \mathbb{R}^{n\times m}$ with $\|\mathbf{A} - \mathbf{A}\mathbf{Z}\mathbf{Z}^T\|_2^2 \leq \frac{\epsilon}{k}\|\mathbf{A} - \mathbf{A}_k\|_F^2$,

$$\min_{\mathbf{X}:rank(\mathbf{X})=k,\mathbf{X}\succeq 0} \|\mathbf{A} - \mathbf{Z}\mathbf{X}\mathbf{Z}^T\|_F^2 \leq (1+c\epsilon)\|\mathbf{A} - \mathbf{A}_k\|_F^2. \tag{2.66}$$

As in Algorithm 1 we can find a near-optimal $\mathbf{X}$ with good probability by further sampling $\mathbf{Z}$ using its leverage scores. W will sample $t_1 = c_1\left(\frac{m\log m/\delta}{\delta^2\epsilon^2}\right)$ rows of $\mathbf{Z}$ by their leverage scores (their norms since $\mathbf{Z}$ is orthonormal) to form $\mathbf{S}_1 \in \mathbb{R}^{n\times t_1}$. By Theorem 39 of [CW13], since $t_1 = \Omega\left(\frac{m\log m/\delta}{\epsilon^2} + \frac{1}{\delta\cdot\epsilon^2}\right)$ if c_1 is set large enough, with probability $\geq 1-\delta/4$ we will have an *affine embedding* of $\mathbf{Z}$. Specifically, letting $\mathbf{B}^* = \arg\min_{\mathbf{B}}\|\mathbf{A} - \mathbf{Z}\mathbf{B}\|_F^2$ and $\mathbf{E}^* = \mathbf{A} - \mathbf{Z}\mathbf{B}^*$, for any $\mathbf{B}$ we have with probability

$\geq 1-\delta/4$:

$$\|\mathbf{S}_1^T\mathbf{A}-\mathbf{S}_1\mathbf{Z}\mathbf{B}\|_F^2+\left(\|\mathbf{E}^*\|_F^2-\|\mathbf{S}_1^T\mathbf{E}^*\|_F^2\right)\in\left[(1-\epsilon)\|\mathbf{A}-\mathbf{Z}\mathbf{B}\|_F^2,(1+\epsilon)\|\mathbf{A}-\mathbf{Z}\mathbf{B}\|_F^2\right]. \tag{2.67}$$

Note that this is similar to the embedding property used in the proof of Theorem 2.5.1 to show that $\mathbf{W}$ computed in Step 6 of Algorithm 1 gave a near-optimal low-rank approximation to $\mathbf{AS}_1$.

By a Markov bound, since $\mathbb{E}\,\|\mathbf{S}_1^T\mathbf{E}^*\|_F^2=\|\mathbf{E}^*\|_F^2$, with probability $\geq 1-\delta/4$,

$$\left|\|\mathbf{E}^*\|_F^2-\|\mathbf{S}_1^T\mathbf{E}^*\|_F^2\right|\leq\frac{4}{\delta}\|\mathbf{E}^*\|_F^2=O(1/\delta)\cdot\|\mathbf{A}-\mathbf{A}_k\|_F^2, \tag{2.68}$$

where the last bound follows from (2.66). Together, (2.67) and (2.68) guarantee that a $(1+\epsilon\delta)$ approximation to the sketched problem gives a $(1+O(\epsilon))$ approximation to the original. That is, with probability $\geq 1-\delta/2$ (union bounding over the probabilities of (2.67) and (2.68) holding) for any PSD $\tilde{\mathbf{X}}$ with $\operatorname{rank}(\tilde{\mathbf{X}})=k$, and

$$\|\mathbf{S}_1^T\mathbf{A}-\mathbf{S}_1\mathbf{Z}\tilde{\mathbf{X}}\mathbf{Z}^T\|_F^2\leq(1+\epsilon\delta)\min_{\mathbf{X}:\operatorname{rank}(\mathbf{X})=k,\mathbf{X}\succeq\mathbf{0}}\|\mathbf{S}_1^T\mathbf{A}-\mathbf{S}_1\mathbf{Z}\mathbf{X}\mathbf{Z}^T\|_F^2 \tag{2.69}$$

we have

$$\|\mathbf{A}-\mathbf{Z}\tilde{\mathbf{X}}\mathbf{Z}^T\|_F^2\leq(1+O(\epsilon))\|\mathbf{A}-\mathbf{A}_k\|_F^2. \tag{2.70}$$

Thus we must show how to efficiently compute $\tilde{\mathbf{X}}$ satisfying 2.69. Following [CW17a], we write $\mathbf{S}_1^T\mathbf{Z}$ in its SVD $\mathbf{S}_1^T\mathbf{Z}=\mathbf{U}_z\mathbf{\Sigma}_z\mathbf{V}_z^T$. Since $\mathbf{S}_1\mathbf{Z}\mathbf{X}\mathbf{Z}^T$ falls in the column span of $\mathbf{S}_1^T\mathbf{Z}$ and the row span of $\mathbf{Z}^T$, we write $\Delta\overset{\text{def}}{=}\|(\mathbf{I}-\mathbf{U}_z\mathbf{U}_z^T)\mathbf{S}_1^T\mathbf{A}\|_F^2+\|\mathbf{U}_z\mathbf{U}_z^T\mathbf{S}_1^T\mathbf{A}(\mathbf{I}-\mathbf{Z}\mathbf{Z}^T)\|_F^2$ and by Pythagorean theorem have for any $\mathbf{X}$:

$$\begin{aligned}\|\mathbf{S}_1^T\mathbf{A}-\mathbf{S}_1\mathbf{Z}\mathbf{X}\mathbf{Z}^T\|_F^2&=\|\mathbf{U}_z\mathbf{U}_z^T\mathbf{S}_1\mathbf{A}\mathbf{Z}\mathbf{Z}^T-\mathbf{U}_z\mathbf{\Sigma}_z\mathbf{V}_z^T\mathbf{X}\mathbf{Z}^T\|_F^2+\Delta\\&=\|\mathbf{U}_z^T\mathbf{S}_1^T\mathbf{A}\mathbf{Z}-\mathbf{\Sigma}_z\mathbf{V}_z^T\mathbf{X}\|_F^2+\Delta\\&=\|\mathbf{\Sigma}_z\mathbf{V}_z^T\left(\mathbf{V}_z\mathbf{\Sigma}_z^{-1}\mathbf{U}_z^T\mathbf{S}_1^T\mathbf{A}\mathbf{Z}-\mathbf{X}\right)\|_F^2+\Delta.\end{aligned} \tag{2.71}$$

Since $\mathbf{S}_1$ is sampled via $\mathbf{Z}$'s leverage scores, and since $t_1=c_1\left(\frac{m\log m/\delta}{\delta^2\epsilon^2}\right)$, if c_1 is set large enough, then with probability $\geq 1-\delta/4$:

$$(1-\delta\epsilon)\mathbf{I}\preceq\mathbf{S}_1^T\mathbf{Z}\preceq(1+\delta\epsilon)\mathbf{I}$$

and so with probability $\geq 1-3\delta/4$, union bounding with the probability (2.71) holds, we have for any $\mathbf{X}$:

$$\|\mathbf{S}_1^T\mathbf{A} - \mathbf{S}_1\mathbf{Z}\mathbf{X}\mathbf{Z}^T\|_F^2 \in (1 \pm \epsilon)\|\mathbf{V}_z^T\mathbf{\Sigma}_z^{-1}\mathbf{U}_z^T\mathbf{S}_1^T\mathbf{A}\mathbf{Z} - \mathbf{X}\|_F^2 + \Delta. \tag{2.72}$$

Finally, following [CW17a], letting $\mathbf{B} = \mathbf{V}_z^T\mathbf{\Sigma}_z^{-1}\mathbf{U}_z^T\mathbf{S}_1^T\mathbf{A}\mathbf{Z}$, we can compute

$$\tilde{\mathbf{X}} = \operatorname*{arg\,min}_{\mathbf{X}|\mathbf{X}\succeq\mathbf{0},\text{rank}(\mathbf{X})=k} \|\mathbf{B} - \mathbf{X}\|_F^2 = \left(\mathbf{B}/2 + \mathbf{B}^T/2\right)_{k,+}$$

where $\mathbf{N}_{k,+}$ has all but the top k *positive* eigenvalues of $\mathbf{N}$ set to 0. By (2.72) and (2.70):

$$\|\mathbf{A} - \mathbf{Z}\tilde{\mathbf{X}}\mathbf{Z}^T\|_F^2 \leq (1 + O(\epsilon))\|\mathbf{A} - \mathbf{A}_k\|_F^2.$$

Thus, if we output $\mathbf{M} = \mathbf{Z}\tilde{\mathbf{X}}^{1/2}$, we have $\|\mathbf{A} - \mathbf{M}\mathbf{M}^T\|_F^2 \leq (1 + O(\epsilon))\|\mathbf{A} - \mathbf{A}_k\|_F^2$, which gives the requires bound.

Overall, the above algorithm requires accessing $O\left(\frac{nm\log m/\delta}{\delta\epsilon^2}\right)$ entries of $\mathbf{A}$ (the entries of $\mathbf{A}\mathbf{S}_1$) and has runtime $\tilde{O}\left(\frac{nm^{\bar{\omega}-1}}{(\delta\epsilon)^{2(\bar{\omega}-1)}}\right)$ giving the lemma. □

Theorem 2.5.7 follows directly from Lemma 2.5.8:

Proof of Theorem 2.5.7. We can obtain $\mathbf{Z}$ with rank $m = \Theta(k/\epsilon)$ that, with probability $\geq 1 - \delta/2$, satisfies:

$$\|\mathbf{A} - \mathbf{A}\mathbf{Z}\mathbf{Z}^T\|_F^2 = O\left(\frac{\epsilon}{k}\right)\|\mathbf{A} - \mathbf{A}_k\|_F^2.$$

by applying Theorem 2.6.3 (proven in Section 2.6) with rank $k' = \Theta(k/\epsilon)$ and error parameter $\epsilon' = \Theta(1)$. This requires accessing $O(\frac{n\cdot k\log^2 n}{\delta^2\cdot\epsilon} + \frac{nk^2}{\delta\cdot\epsilon^2})$ entries of $\mathbf{A}$, along with $\tilde{O}\left(\frac{nk^{\bar{\omega}}}{\delta\epsilon^{\bar{\omega}}} + \frac{nk}{\delta^2\cdot\epsilon} + (\sqrt{n}k^{\bar{\omega}-1} + k^{\bar{\omega}+1})\cdot\text{poly}(1/\epsilon, 1/\delta)\right)$ runtime.

Applying Lemma 2.5.8 with error $\Theta(\epsilon)$ and failure probability $\delta/2$ then gives the theorem via a union bound. This requires $O\left(\frac{nk\log k/(\delta\epsilon)}{\delta^2\epsilon^3}\right)$ accesses to $\mathbf{A}$, along with $\tilde{O}\left(\frac{nk^{\bar{\omega}-1}}{(\delta^2\cdot\epsilon^3)^{(\bar{\omega}-1)}}\right)$ runtime. Overall, the algorithm requires $\tilde{O}\left(\frac{nk}{\delta^2\cdot\epsilon^3} + \frac{nk^2}{\delta\cdot\epsilon^2}\right)$ accesses to $\mathbf{A}$ and $\tilde{O}\left(\frac{nk^{\bar{\omega}}}{\delta\epsilon^{\bar{\omega}}} + \frac{nk^{\bar{\omega}-1}}{(\delta^2\cdot\epsilon^3)^{(\bar{\omega}-1)}} + (\sqrt{n}k^{\bar{\omega}-1} + k^{\bar{\omega}+1})\cdot\text{poly}(1/\epsilon, 1/\delta)\right)$ runtime. □

2.6 Spectral Norm Error Bounds

In this section we show how to modify Algorithm 1 to solve the following spectral norm low-rank approximation problem for PSD $\mathbf{A}$:

Problem 2.6.1 (Near-Optimal Low-Rank Approximation – Spectral Norm)**.** *Given any* $\mathbf{A} \in \mathbb{R}^{n\times d}$*, rank parameter* $k \in \mathbb{Z}^{\geq 1}$*, and accuracy parameter* $\epsilon \geq 0$*, output* $\mathbf{M} \in \mathbb{R}^{n\times k}$*,* $\mathbf{N} \in \mathbb{R}^{d\times k}$ *such that, letting* $\mathbf{B} = \mathbf{M}\mathbf{N}^T$*,*

$$\|\mathbf{A} - \mathbf{B}\|_2^2 \leq (1+\epsilon)\|\mathbf{A} - \mathbf{A}_k\|_2^2 + \frac{\epsilon}{k}\|\mathbf{A} - \mathbf{A}_k\|_F^2, \tag{2.73}$$

where $\mathbf{A}_k = \arg\min_{\text{rank-}k\ \mathbf{B}} \|\mathbf{A} - \mathbf{B}\|_2 = \arg\min_{\text{rank-}k\ \mathbf{B}} \|\mathbf{A} - \mathbf{B}\|_F$*.*

This can be significantly stronger than the Frobenius guarantee of Problem 2.1.1 when $\|\mathbf{A} - \mathbf{A}_k\|_F^2$ is large, and, for example is critical in our application to ridge regression, discussed in Section 2.6.3.

2.6.1 Algorithmic Approach

It is not hard to see that since additive error in the Frobenius norm upper bounds additive error in the spectral norm (see e.g. Theorem 3.4 of [Gu14]) that for $\mathbf{B}$ satisfying the Frobenius norm guarantee of Problem 2.1.1, $\|\mathbf{A} - \mathbf{B}\|_F^2 \leq (1+\epsilon)\|\mathbf{A} - \mathbf{A}_k\|_F^2$, we immediately have the spectral bound $\|\mathbf{A}-\mathbf{B}\|_2^2 \leq \|\mathbf{A}-\mathbf{A}_k\|_2^2+\epsilon\|\mathbf{A}-\mathbf{A}_k\|_F^2$. Thus, we can solve Problem 2.6.1 simply by running Algorithm 1 with error parameter ϵ/k. However, this approach is suboptimal. Applying Theorem 2.5.1, our query complexity would be $\Omega\left(\frac{nk^{3.5}\log^2 n}{\delta^2\cdot\epsilon^{2.5}}\right)$. We improve this k dependence significantly in Algorithm 2. Since Problem 2.6.1 is often applied (see for example Section 2.6.3) with $k' = k/\epsilon$ and $\epsilon = \Theta(1)$ to give $\|\mathbf{A} - \mathbf{B}\|_2^2 \leq O\left(\frac{\epsilon}{k}\|\mathbf{A} - \mathbf{A}_k\|_F^2\right)$, optimizing the dependence on k is especially important.

We first give an extension of the projection-cost-preserving sketch result of Lemma 2.2.12 to the spectral norm case. This lemma provides the column sampling analog to Lemma 2.4.3.

Lemma 2.6.2 (Spectral Norm PCP)**.** *For any* $\mathbf{A} \in \mathbb{R}^{n\times d}$*, for all* $i \in [d]$*, let* $\tilde{\tau}_i^k \geq \tau_i^k(\mathbf{A})$ *be an overestimate for the* i^{th} *rank-k ridge leverage score. Let* $p_i = \frac{\tilde{\tau}_i^k}{\sum_i \tilde{\tau}_i^k}$ *and* $t = \frac{c\log(k/\delta)}{\epsilon^2}\sum_i \tilde{\tau}_i^k$ *for any* $\epsilon, \delta \in (0, 1/2]$ *and sufficiently large constant c. Construct* $\mathbf{C}$ *by sampling t columns of* $\mathbf{A}$*, each set to* $\frac{1}{\sqrt{tp_i}}\mathbf{a}_i$ *with probability* p_i*. With probability* $\geq 1-\delta$*,* $\mathbf{C}$ *is an* (ϵ, α, k)*-spectral PCP of* $\mathbf{A}$ *for* $\alpha = \frac{\epsilon}{k}\|\mathbf{A} - \mathbf{A}_k\|_F^2$ *(Definition 2.4.2). That is, for any orthogonal projection* $\mathbf{P} \in \mathbb{R}^{n\times n}$*,*

$$(1-\epsilon)\|\mathbf{A} - \mathbf{P}\mathbf{A}\|_2^2 - \frac{\epsilon}{k}\|\mathbf{A} - \mathbf{A}_k\|_F^2 \leq \|\mathbf{C} - \mathbf{P}\mathbf{C}\|_2^2 \leq (1+\epsilon)\|\mathbf{A} - \mathbf{P}\mathbf{A}\|_2^2 + \frac{\epsilon}{k}\|\mathbf{A} - \mathbf{A}_k\|_F^2.$$

Proof. The lemma follows rather directly from Corollary 2.2.11. If c is set large enough, with probability $\geq 1-\delta$, sampling by the rank-k ridge leverage scores gives

$\mathbf{C}$ satisfying:

$$(1-\epsilon)\mathbf{C}\mathbf{C}^T - \frac{\epsilon}{k}\|\mathbf{A}-\mathbf{A}_k\|_F^2\mathbf{I} \preceq \mathbf{A}\mathbf{A}^T \preceq (1+\epsilon)\mathbf{C}\mathbf{C}^T + \frac{\epsilon}{k}\|\mathbf{A}-\mathbf{A}_k\|_F^2\mathbf{I}. \tag{2.74}$$

We can write for any $\mathbf{M}$, $\|\mathbf{M}\|_2^2 = \max_{\mathbf{x}:\|\mathbf{x}\|_2^2=1}\mathbf{x}^T\mathbf{M}\mathbf{x}$. When $\|\mathbf{x}\|_2^2 = 1$, $\|(\mathbf{I}-\mathbf{P})\mathbf{x}\|_2^2 \leq 1$ so by (2.74) we have for any unit norm $\mathbf{x}$:

$$\begin{aligned}\mathbf{x}^T(\mathbf{I}-\mathbf{P})\mathbf{C}\mathbf{C}^T(\mathbf{I}-\mathbf{P})\mathbf{x} &\leq \mathbf{x}^T(\mathbf{I}-\mathbf{P})\mathbf{A}\mathbf{A}^T(\mathbf{I}-\mathbf{P})\mathbf{x} + \frac{\epsilon}{k}\|\mathbf{A}-\mathbf{A}_k\|_F^2\\ &\leq \|\mathbf{A}-\mathbf{P}\mathbf{A}\|_2^2 + \frac{\epsilon}{k}\|\mathbf{A}-\mathbf{A}_k\|_F^2,\end{aligned}$$

which gives $\|\mathbf{C}-\mathbf{P}\mathbf{C}\|_2^2 \leq \frac{1}{1-\epsilon}\|\mathbf{A}-\mathbf{P}\mathbf{A}\|_2^2 + \frac{\epsilon}{(1-\epsilon)k}\|\mathbf{A}-\mathbf{A}_k\|_F^2$. Similarly we have:

$$\begin{aligned}\mathbf{x}^T(\mathbf{I}-\mathbf{P})\mathbf{A}\mathbf{A}^T(\mathbf{I}-\mathbf{P})x &\leq (1+\epsilon)\mathbf{x}^T(\mathbf{I}-\mathbf{P})\mathbf{C}\mathbf{C}^T(\mathbf{I}-\mathbf{P})\mathbf{x} + \frac{\epsilon}{k}\|\mathbf{A}-\mathbf{A}_k\|_F^2\\ &\leq (1+\epsilon)\|\mathbf{C}-\mathbf{P}\mathbf{C}\|_2^2 + \frac{\epsilon}{k}\|\mathbf{A}-\mathbf{A}_k\|_F^2,\end{aligned}$$

which gives $\frac{1}{1+\epsilon}\|\mathbf{A}-\mathbf{P}\mathbf{A}\|_2^2 - \frac{\epsilon}{(1+\epsilon)k}\|\mathbf{A}-\mathbf{A}_k\|_F^2 \leq \|\mathbf{C}-\mathbf{P}\mathbf{C}\|_2^2$. The lemma follows from combining these upper and lower bounds after adjusting constant factors on ϵ (by making c large enough). □

2.6.2 Basic Algorithm

We now use Lemma 2.6.2, along with its row sampling counterpart, Lemma 2.4.3, to give an algorithm (Algorithm 2) for computing a near-optimal spectral norm low-rank approximation to $\mathbf{A}$.

In Steps 1-3 we sample both rows and columns of $\mathbf{A}$ via the rank-$\Theta(k/\epsilon^2)$ ridge leverage scores of $\mathbf{A}^{1/2}$, ensuring with high probability that $\mathbf{A}\mathbf{S}_1$ is an (ϵ,α,k)-spectral PCP of $\mathbf{A}$ with $\alpha = \frac{\epsilon}{k}\|\mathbf{A}-\mathbf{A}_k\|_F^2$ and that $\tilde{\mathbf{A}}$ is in turn an (ϵ,α,k)-spectral row PCP of $\mathbf{A}\mathbf{S}_1$. Thus, if we compute (using an input sparsity time algorithm) a span $\mathbf{Z}$ that gives a near-optimal spectral norm low-rank approximation to $\tilde{\mathbf{A}}$ (Step 3), this span will also be nearly optimal for $\mathbf{A}\mathbf{S}_1$. Since we cannot afford to fully read $\mathbf{A}\mathbf{S}_1$, we approximately project it to $\mathbf{Z}$ by further sampling its columns using $\mathbf{Z}$'s leverage scores (Step 4). We use leverage score sampling again in Step 5 to approximately project $\mathbf{A}$ to the span of the result. This yields our final approximation, using the fact that $\mathbf{A}\mathbf{S}_1$ is a spectral PCP for $\mathbf{A}$.

```
Algorithm 2  PSD Low-Rank Approximation – Spectral Error

1. Let $k_1 = \lceil ck/\epsilon^2 \rceil$. Compute, with probability $\geq 1 - \delta/5$ using the algorithm of Lemma 2.2.17, $\tilde{\tau}_i^{k_1}(\mathbf{A}^{1/2})$ satisfying for all $i \in [n]$:
    • $\tau_i^{k_1}(\mathbf{A}^{1/2}) \leq \tilde{\tau}_i^{k_1}(\mathbf{A}^{1/2}) \leq 3\tau_i^{k_1}(\mathbf{A}^{1/2})$.

2. Set $\ell_i^{(1)} = 4\epsilon\sqrt{\frac{n}{k}}\tilde{\tau}_i^{k_1}(\mathbf{A}^{1/2})$. Set $p_i^{(1)} = \frac{\ell_i^{(1)}}{\sum_i \ell_i^{(1)}}$ and $t_1 = \frac{c_1 \log n}{\delta \cdot \epsilon^2} \sum_i \ell_i^{(1)}$. Sample $\mathbf{S}_1, \mathbf{S}_2 \in \mathbb{R}^{n \times t_1}$ each whose $j^{th}$ column is set to $\frac{1}{\sqrt{tp_i^{(1)}}}\mathbf{e}_i$ with probability $p_i^{(1)}$.

3. Let $\tilde{\mathbf{A}} = \mathbf{S}_2^T \mathbf{A} \mathbf{S}_1$, and use an input sparsity time algorithm to compute orthonormal $\mathbf{Z} \in \mathbb{R}^{t_1 \times k}$ satisfying with probability $\geq 1 - \delta/5$ both:
    • $\|\tilde{\mathbf{A}} - \tilde{\mathbf{A}}\mathbf{Z}\mathbf{Z}^T\|_F^2 \leq (1+\delta)\|\tilde{\mathbf{A}} - \tilde{\mathbf{A}}_k\|_F^2$.
    • $\|\tilde{\mathbf{A}} - \tilde{\mathbf{A}}\mathbf{Z}\mathbf{Z}^T\|_2^2 \leq (1+\epsilon)\|\tilde{\mathbf{A}} - \tilde{\mathbf{A}}_k\|_2^2 + \frac{\epsilon}{k}\|\tilde{\mathbf{A}} - \tilde{\mathbf{A}}_k\|_F^2$.

4. Let $t_3 = c_3\left(k \log k + \frac{k^2}{\delta \cdot \epsilon}\right)$, set $p_i^{(3)} = \frac{\|\mathbf{z}_i\|_2^2}{\|\mathbf{Z}\|_F^2}$, and sample $\mathbf{S}_3 \in \mathbb{R}^{t_1 \times t_3}$ where the $j^{th}$ column is set to $\frac{1}{\sqrt{t_3 p_i^{(3)}}}\mathbf{e}_i$ with probability $p_i^{(3)}$. Solve:
    $$\mathbf{W} = \arg\min_{\mathbf{W} \in \mathbb{R}^{n \times k}} \|\mathbf{A}\mathbf{S}_1\mathbf{S}_3 - \mathbf{W}\mathbf{Z}^T\mathbf{S}_3\|_F^2.$$

5. Compute an orthonormal basis $\mathbf{M} \in \mathbb{R}^{n \times k}$ for the column span of $\mathbf{W}$. Let $t_4 = c_4\left(k \log k + \frac{k^2}{\delta \cdot \epsilon}\right)$, set $p_i^{(4)} = \frac{\|\mathbf{m}_i\|_2^2}{\|\mathbf{M}\|_F^2}$, and sample $\mathbf{S}_4 \in \mathbb{R}^{n \times t_4}$ where the $j^{th}$ column is set to $\frac{1}{\sqrt{t_4 p_i^{(4)}}}\mathbf{e}_i$ with probability $p_i^{(4)}$. Solve:
    $$\mathbf{N} = \arg\min_{\mathbf{N} \in \mathbb{R}^{n \times k}} \|\mathbf{S}_4^T\mathbf{M}\mathbf{N}^T - \mathbf{S}_4^T\mathbf{A}\|_F^2.$$

6. Return $\mathbf{M}, \mathbf{N} \in \mathbb{R}^{n \times k}$.
```

Theorem 2.6.3 (Sublinear Time Low-Rank Approximation – Spectral Norm Error)**.** *Given any PSD* $\mathbf{A} \in \mathbb{R}^{n \times n}$*, for sufficiently large constants* c, c_1, c_2, c_3, c_4*, for any* $k \in \mathbb{Z}^{\geq 1}$ *and* $\epsilon, \delta \in (0, 1]$*, Algorithm 2 accesses* $O(\frac{n \cdot k \log^2 n}{\delta^2 \cdot \epsilon^6} + \frac{nk^2}{\delta \cdot \epsilon})$ *entries of* $\mathbf{A}$*, runs in*

$$\tilde{O}\left(\frac{nk^{\bar{\omega}}}{\delta\epsilon} + \frac{nk}{\delta^2 \cdot \epsilon^6} + (\sqrt{n}k^{\bar{\omega}-1} + k^{\bar{\omega}+1}) \cdot \text{poly}(1/\epsilon, 1/\delta)\right)$$

time and with probability $\geq 1-\delta$ *outputs* $\mathbf{M}, \mathbf{N} \in \mathbb{R}^{n \times k}$ *with:*

$$\|\mathbf{A}-\mathbf{M}\mathbf{N}^T\|_2^2 \leq (1+\epsilon)\|\mathbf{A}-\mathbf{A}_k\|_2^2 + \frac{\epsilon}{k}\|\mathbf{A}-\mathbf{A}_k\|_F^2.$$

Proof. We begin with the following claim, which shows that $\mathbf{A}\mathbf{S}_1$ and $\tilde{\mathbf{A}}$ satisfy both spectral and Frobenius norm PCP bounds.

Claim 2.6.4. *With probability* $\geq 1 - 3\delta/5$ *we simultaneously have:*

- $\mathbf{A}\mathbf{S}_1$ *is an* (ϵ, α, k)*-spectral PCP of* $\mathbf{A}$ *with* $\alpha = \frac{\epsilon}{k}\|\mathbf{A}-\mathbf{A}_k\|_F^2$.
- $\tilde{\mathbf{A}}$ *is an* (ϵ, α, k)*-spectral row PCP of* $\mathbf{A}\mathbf{S}_1$ *with* $\alpha = \frac{\epsilon}{k}\|\mathbf{A}-\mathbf{A}_k\|_F^2$.
- $\mathbf{A}\mathbf{S}_1$ *is an* (ϵ, k)*-PCP of* $\mathbf{A}$.
- $\tilde{\mathbf{A}}$ *is an* $(\epsilon, c_5/\delta, k)$*-PCP for* $\mathbf{AS_1}$ *for some constant* c_5.

Proof. $\ell_i^{(1)} = 4\epsilon\sqrt{\frac{n}{k}}\tilde{\tau}_i^{k_1}(\mathbf{A}^{1/2})$, which by Lemma 2.3.1 is within a constant factor of upper bounding the rank $k_1 = \lceil ck/\epsilon^2 \rceil$ ridge leverage scores of $\mathbf{A}$. As long as $c \geq 1$, these scores upper bound the rank-k ridge leverage scores. So by Lemma 2.6.2, if c_1 is set large enough, with probability $\geq 1-\delta/5$, $\mathbf{A}\mathbf{S}_1$ is an (ϵ, α, k)-spectral PCP of $\mathbf{A}$ with $\alpha = \frac{\epsilon}{k}\|\mathbf{A}-\mathbf{A}_k\|_F^2$. Additionally, by Lemma 2.4.3, with probability $\geq 1-\delta/5$, $\tilde{\mathbf{A}}$ is an (ϵ, α, k)-spectral row PCP of $\mathbf{A}\mathbf{S}_1$ with $\alpha = \frac{\epsilon}{k}\|\mathbf{A}-\mathbf{A}_k\|_F^2$.

By Lemma 2.4.1, since $t_1 = \frac{c_1 \log n}{\delta \cdot \epsilon^2}\sum_i \ell_i^{(1)}$, if c_1 is set large enough, with probability $\geq 1-\delta/5$ we have simultaneously that $\mathbf{A}\mathbf{S}_1$ is an (ϵ, k_1)-PCP of $\mathbf{A}$ and that $\tilde{\mathbf{A}}$ is an $(\epsilon, c_5/\delta, k_1)$-PCP for $\mathbf{AS_1}$ for some constant c_5. These bounds immediately give that $\mathbf{A}\mathbf{S}_1$ is an (ϵ, k)-PCP for $\mathbf{A}$ and $\tilde{\mathbf{A}}$ is an $(\epsilon, c_6/\delta, k_1)$-PCP for $\mathbf{AS_1}$. The claim follows via a union bound. □

Assuming the bounds of Claim 2.6.4 hold, from bound (3) and (4) we have:

$$\|\tilde{\mathbf{A}}-\tilde{\mathbf{A}}_k\|_F^2 \leq c_6\|\mathbf{A}-\mathbf{A}_k\|_F^2 \tag{2.75}$$

for some constant c_6. Overall, with probability $\geq 1-3\delta/5$ both the above Frobenius norm PCP bounds as well as the spectral PCP bounds hold. By (2.75), assuming that $\tilde{\mathbf{A}}$ is in fact an (ϵ, α, k)-spectral PCP of $\mathbf{A}\mathbf{S}_1$ with $\alpha = \frac{\epsilon}{k}\|\mathbf{A}-\mathbf{A}_k\|_F^2$, for $\mathbf{Z}$ computed in Step 3 of Algorithm 2 we have:

$$\begin{aligned}
\|\mathbf{AS}_1 - \mathbf{AS}_1\mathbf{ZZ}^T\|_2^2 &\le \frac{(1+\epsilon)}{(1-\epsilon)}\left(\|\tilde{\mathbf{A}} - \tilde{\mathbf{A}}_k\|_2^2 + \frac{\epsilon}{k}\|\tilde{\mathbf{A}} - \tilde{\mathbf{A}}_k\|_F^2\right) + \frac{\epsilon}{k(1-\epsilon)}\|\mathbf{A} - \mathbf{A}_k\|_F^2 \\
&\le (1+3\epsilon)\|\tilde{\mathbf{A}} - \tilde{\mathbf{A}}_k\|_2^2 + \frac{(3c_6+2)\epsilon}{k}\|\mathbf{A} - \mathbf{A}_k\|_F^2 \\
&\le (1+3\epsilon)(1+\epsilon)\|\mathbf{AS}_1 - (\mathbf{AS}_1)_k\|_2^2 + \frac{(3c_6+2+1)\epsilon}{k}\|\mathbf{A} - \mathbf{A}_k\|_F^2,
\end{aligned} \tag{2.76}$$

where we use that $\epsilon \le 1/2$.

Now, by the approximate regression result of Lemma 2.2.14, since $\mathbf{S}_3$ is sampled by the leverage scores of $\mathbf{Z}$, and since $t_3 = c_3\left(k\log k + \frac{k}{\delta\cdot\epsilon'}\right)$ for $\epsilon' = \epsilon/k$, if c_3 is set large enough, for $\mathbf{W}$ computed in Step 4, with probability $\ge 1-\delta/5$:

$$\|\mathbf{AS}_1 - \mathbf{WZ}^T\|_F^2 \le \left(1+\frac{\epsilon}{k}\right)\|\mathbf{AS}_1 - \mathbf{AS}_1\mathbf{ZZ}^T\|_F^2. \tag{2.77}$$

This Frobenius norm bound also implies a spectral norm bound. Specifically, we can write:

$$\|\mathbf{AS}_1 - \mathbf{WZ}^T\|_F^2 = \|\mathbf{AS}_1\mathbf{ZZ}^T - \mathbf{WZ}^T\|_F^2 + \|\mathbf{AS}_1(\mathbf{I} - \mathbf{ZZ}^T)\|_F^2$$

so by (2.77) we must have $\|\mathbf{AS}_1\mathbf{ZZ}^T - \mathbf{WZ}^T\|_F^2 \le \frac{\epsilon}{k}\|\mathbf{AS}_1(\mathbf{I} - \mathbf{ZZ}^T)\|_F^2$. This gives:

$$\begin{aligned}
\|\mathbf{AS}_1 - \mathbf{WZ}^T\|_2^2 &\le \|\mathbf{AS}_1\mathbf{ZZ}^T - \mathbf{WZ}^T\|_2^2 + \|\mathbf{AS}_1(\mathbf{I} - \mathbf{ZZ}^T)\|_2^2 \\
&\le \|\mathbf{AS}_1(\mathbf{I} - \mathbf{ZZ}^T)\|_2^2 + \|\mathbf{AS}_1\mathbf{ZZ}^T - \mathbf{WZ}^T\|_F^2 \\
&\le \|\mathbf{AS}_1(\mathbf{I} - \mathbf{ZZ}^T)\|_2^2 + \frac{\epsilon}{k}\|\mathbf{AS}_1(\mathbf{I} - \mathbf{ZZ}^T)\|_F^2.
\end{aligned} \tag{2.78}$$

Note that by the Frobenius norm guarantee required in Step 3, (2.75), and the fact that $\tilde{\mathbf{A}}$ is an $(\epsilon, c_5/\delta, k)$ PCP for $\mathbf{AS}_1$,

$$\|\mathbf{AS}_1 - \mathbf{AS}_1\mathbf{ZZ}^T\|_F^2 = O(\|\mathbf{A} - \mathbf{A}_k\|_F^2).$$

Combined with (2.75) and (2.78) the above implies that for $\mathbf{M}$ spanning the columns of $\mathbf{W}$,

$$\begin{aligned}
\|\mathbf{AS}_1 - \mathbf{MM}^T\mathbf{AS}_1\|_2^2 &\le \|\mathbf{AS}_1 - \mathbf{WZ}^T\|_2^2 \\
&\le (1+O(\epsilon))\|\mathbf{AS}_1 - (\mathbf{AS}_1)_k\|_2^2 + O\left(\frac{\epsilon}{k}\right)\|\mathbf{A} - \mathbf{A}_k\|_F^2.
\end{aligned}$$

Assuming that $\mathbf{AS}_1$ is in fact an (ϵ, α, k)-spectral PCP for $\mathbf{A}$ with $\alpha = \frac{\epsilon}{k}\|\mathbf{A} - \mathbf{A}_k\|_F^2$ we also have

$$\|\mathbf{A} - \mathbf{M}\mathbf{M}^T\mathbf{A}\|_2^2 \leq (1 + O(\epsilon))\|\mathbf{A} - \mathbf{A}_k\|_2^2 + O\left(\frac{\epsilon}{k}\right)\|\mathbf{A} - \mathbf{A}_k\|_F^2. \tag{2.79}$$

Union bounding over the probability that Claim 2.6.4 holds ($\geq 1 - 3\delta/5$) and that (2.77) holds ($\geq 1 - \delta/5$), (2.79) holds with probability $\geq 1 - 4\delta/5$.

Finally, we apply an identical approximate regression argument for $\mathbf{N}$ computed in Step 5 to show that, with probability $\geq 1 - \delta/5$, assuming that (2.79) holds,

$$\begin{aligned}\|\mathbf{A} - \mathbf{M}\mathbf{N}^T\|_2^2 &\leq (1 + \epsilon)\|\mathbf{A} - \mathbf{M}\mathbf{M}^T\mathbf{A}\|_2^2 + \frac{\epsilon}{k}\|\mathbf{A} - \mathbf{M}\mathbf{M}^T\mathbf{A}\|_F^2 \\ &= (1 + O(\epsilon))\|\mathbf{A} - \mathbf{A}_k\|_2^2 + O\left(\frac{\epsilon}{k}\right)\|\mathbf{A} - \mathbf{A}_k\|_F^2.\end{aligned}$$

By a union bound, this holds with probability $\geq 1 - \delta$. Adjusting constants on ϵ by making c, c_1, c_2, c_3, c_4 large enough yields the final bound.

Runtime and Sample Complexity

It just remains to discuss runtime and sample complexity.

Step 1 requires accessing $O\left(\frac{nk\log(k/(\epsilon\delta))}{\epsilon^2}\right)$ entries of $\mathbf{A}$ by Lemma 2.2.17. Constructing $\mathbf{S}_2^T\mathbf{A}\mathbf{S}_1$ in Step 3 requires reading $t_1^2 = O\left(\frac{nk\log^2 n}{\delta^2\cdot\epsilon^6}\right)$ entries of $\mathbf{A}$. Constructing $\mathbf{A}\mathbf{S}_1\mathbf{S}_3$ in Step 4 and $\mathbf{A}\mathbf{S}_4$ in Step 5 both require reading $O\left(nk\log k + \frac{nk^2}{\delta\cdot\epsilon}\right)$ entries.

For runtime, computing $\mathbf{Z}$ in Step 3 requires $O(\text{nnz}(\tilde{\mathbf{A}}))+\tilde{O}(\sqrt{n}k^{\bar{\omega}-1}\cdot\text{poly}(1/\epsilon, 1/\delta))$ time using an input sparsity time algorithm (e.g., by Theorem 27 of [CEM+15] or using input sparsity time ridge leverage score sampling [CMM17] in conjunction with the spectral PCP result of Lemma 2.6.2). Computing $\mathbf{M}$ in Step 4 and $\mathbf{N}$ in Step 5 both require computing the pseudoinverse of a $k \times O\left(k\log k + \frac{k^2}{\delta\cdot\epsilon}\right)$ matrix, which takes $\tilde{O}(k^{\bar{\omega}+1}\,\text{poly}(1/\epsilon, 1/\delta))$ time, and then applying this to an $n \times O\left(k\log k + \frac{k^2}{\delta\epsilon}\right)$ matrix, which requires $\tilde{O}\left(\frac{nk^{\bar{\omega}}}{\delta\epsilon}\right)$ time. □

2.6.3 Sublinear Time Ridge Regression

We now demonstrate how Theorem 2.6.3 can be leveraged to give a sublinear time, relative error algorithm for approximately solving the following ridge regression problem for PSD $\mathbf{A}$:

Problem 2.6.5 (Ridge Regression)**.** *Given any* $\mathbf{A} \in \mathbb{R}^{n\times d}$, $\mathbf{y} \in \mathbb{R}^n$, *regularization parameter* $\lambda \geq 0$, *and error parameter* $\epsilon \geq 0$, *compute* $\tilde{\mathbf{x}}$ *satisfying:*

$$\|\mathbf{A}\tilde{\mathbf{x}} - \mathbf{y}\|_2^2 + \lambda\|\tilde{\mathbf{x}}\|_2^2 \leq (1+\epsilon) \cdot \min_{\mathbf{x}\in\mathbb{R}^n} \|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2 + \lambda\|\mathbf{x}\|_2^2. \tag{2.80}$$

We begin with a lemma showing that for PSD $\mathbf{A}$, any approximation to $\mathbf{A}$ with small spectral norm error can be used to approximately solve (2.80) up to relative error.

Lemma 2.6.6 (Ridge Regression via Spectral Norm Low-Rank Approximation)**.** *For any PSD* $\mathbf{A} \in \mathbb{R}^{n\times n}$, $\mathbf{y} \in \mathbb{R}^n$, *regularization parameter* $\lambda \geq 0$ *and* $\mathbf{B}$ *with* $\|\mathbf{A} - \mathbf{B}\|_2^2 \leq \epsilon^2\lambda$, *let* $\tilde{\mathbf{x}} \in \mathbb{R}^n$ *be any vector satisfying:*

$$\|\mathbf{B}\tilde{\mathbf{x}} - \mathbf{y}\|_2^2 + \lambda\|\tilde{\mathbf{x}}\|_2^2 \leq (1+\alpha) \cdot \min_{\mathbf{x}\in\mathbb{R}^n} \|\mathbf{B}\mathbf{x} - \mathbf{y}\|_2^2 + \lambda\|\mathbf{x}\|_2^2.$$

Then:

$$\|\mathbf{A}\tilde{\mathbf{x}} - \mathbf{y}\|_2^2 + \lambda\|\tilde{\mathbf{x}}\|_2^2 \leq (1+\alpha)(1+5\epsilon) \min_{\mathbf{x}\in\mathbb{R}^n} \|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2 + \lambda\|\mathbf{x}\|_2^2.$$

Proof. For any $\mathbf{x} \in \mathbb{R}^n$ we have:

$$\|\mathbf{B}\mathbf{x} - \mathbf{y}\|_2^2 + \lambda\|\mathbf{x}\|_2^2 = \|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2 + \|(\mathbf{B} - \mathbf{A})\mathbf{x}\|_2^2 + 2\mathbf{x}^T(\mathbf{B} - \mathbf{A})^T(\mathbf{A}\mathbf{x} - \mathbf{y}) + \lambda\|\mathbf{x}\|_2^2.$$

By the requirement that $\|\mathbf{A} - \mathbf{B}\|_2^2 \leq \epsilon^2\lambda$, we have $\|(\mathbf{B} - \mathbf{A})\mathbf{x}\|_2^2 \leq \epsilon^2\lambda\|\mathbf{x}\|_2^2$ and further:

$$\begin{aligned}
|2\mathbf{x}^T(\mathbf{B} - \mathbf{A})^T(\mathbf{A}\mathbf{x} - \mathbf{y})| &\leq 2\|(\mathbf{A} - \mathbf{B})\mathbf{x}\|_2\|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2 \\
&\leq 2\epsilon\sqrt{\lambda}\|\mathbf{x}\|_2\|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2 \\
&\leq \epsilon\left(\|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2 + \lambda\|\mathbf{x}\|_2^2\right).
\end{aligned}$$

So for any $\mathbf{x} \in \mathbb{R}^n$,

$$\|\mathbf{B}\mathbf{x} - \mathbf{y}\|_2^2 + \lambda\|\mathbf{x}\|_2^2 \in [(1-2\epsilon)\left(\|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2 + \lambda\|\mathbf{x}\|_2^2\right), (1+2\epsilon)\left(\|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2 + \lambda\|\mathbf{x}\|_2^2\right)],$$

which gives the lemma since any nearly optimal $\tilde{\mathbf{x}}$ for the ridge regression problem on $\mathbf{B}$ will also be nearly optimal for $\mathbf{A}$. □

Combining Lemma 2.6.6 with Theorem 2.6.3 gives the following:

Theorem 2.6.7 (Sublinear Time Ridge Regression)**.** *Given any PSD* $\mathbf{A} \in \mathbb{R}^{n \times n}$*, regularization parameter* $\lambda \geq 0$*,* $\mathbf{y} \in \mathbb{R}^n$*,* $\epsilon, \delta \in (0, 1/2]$ *and upper bound* $\tilde{s}_\lambda$ *on the statistical dimension* $s_\lambda \overset{\text{def}}{=} \operatorname{tr}((\mathbf{A}^2 + \lambda \mathbf{I})^{-1} \mathbf{A}^2)$*, there is an algorithm accessing* $\tilde{O}\left(\frac{n\tilde{s}_\lambda^2}{\delta \cdot \epsilon^4} + \frac{n\tilde{s}_\lambda}{\delta^2 \cdot \epsilon^2}\right)$ *entries of* $\mathbf{A}$ *and running in* $\tilde{O}\left(\frac{n\tilde{s}_\lambda^{\bar{\omega}}}{\delta \cdot \epsilon^{2\bar{\omega}}} + \frac{n\tilde{s}_\lambda}{\delta^2 \epsilon^2} + (\sqrt{n} s_\lambda^{\bar{\omega}-1} + s_\lambda^{\bar{\omega}+1}) \cdot \operatorname{poly}(1/\epsilon, 1/\delta)\right)$ *time, that outputs* $\tilde{\mathbf{x}}$ *satisfying with probability* $\geq 1 - \delta$*:*

$$\|\mathbf{A}\tilde{\mathbf{x}} - \mathbf{y}\|_2^2 + \lambda \|\tilde{\mathbf{x}}\|_2^2 \leq (1 + \epsilon) \cdot \min_{\mathbf{x} \in \mathbb{R}^n} \|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2 + \lambda \|\mathbf{x}\|_2^2.$$

When $\tilde{s}_\lambda \ll n$ as is often the case, the above significantly improves upon state-of-the-art input sparsity time runtimes for general matrices [ACW16].

Proof. Let $k = \frac{c\tilde{s}_\lambda}{\epsilon^2}$ for sufficiently large constant c. We have:

$$s_\lambda = \sum_{i=1}^{n} \frac{\sigma_i^2(\mathbf{A})}{\sigma_i^2(\mathbf{A}) + \lambda} \geq \sum_{i : \sigma_i^2(\mathbf{A}) \geq \epsilon^2 \lambda} \frac{\sigma_i^2(\mathbf{A})}{(1 + 1/\epsilon^2)\sigma_i^2(\mathbf{A})} \geq \frac{\epsilon^2}{2} \cdot |\{i : \sigma_i^2(\mathbf{A}) \geq \epsilon^2 \lambda\}|.$$

So $|\{i : \sigma_i^2(\mathbf{A}) \geq \epsilon^2 \lambda\}| \leq \frac{2s_\lambda}{\epsilon^2}$ and for large enough c and $k = \frac{c\tilde{s}_\lambda}{\epsilon^2} \geq \frac{cs_\lambda}{\epsilon^2}$ we have $\|\mathbf{A} - \mathbf{A}_k\|_2^2 \leq \frac{\epsilon^2 \lambda}{2}$ and can run Algorithm 2 with error parameter $\epsilon = \Theta(1)$ and failure probability $1 - \delta$ to find $\mathbf{M}, \mathbf{N} \in \mathbb{R}^{n \times k}$ with $\|\mathbf{A} - \mathbf{M}\mathbf{N}^T\|_2^2 \leq \epsilon^2 \lambda$. We can then apply Lemma 2.6.6 – solving $\tilde{\mathbf{x}} = \min_{\mathbf{x} \in \mathbb{R}^n} \|\mathbf{M}\mathbf{N}^T\mathbf{x} - \mathbf{y}\|_2^2 + \lambda \|\mathbf{x}\|_2^2$ exactly using an SVD in $O(nk^{\bar{\omega}-1})$ time. $\tilde{\mathbf{x}}$ will be a $(1 + O(\epsilon))$ approximate solution for $\mathbf{A}$, which, after adjusting constants on ϵ, gives the lemma. The runtime follows from Theorem 2.6.3 with $k = c\tilde{s}_\lambda/\epsilon^2$ and $\epsilon' = \Theta(1)$. The $O(nk^{\bar{\omega}-1})$ regression cost is dominated by the cost of computing the low-rank approximation. □

Note that Theorem 2.6.7 ensures that if the $k \geq \frac{cs_\lambda}{\epsilon^2}$ for some constant c, $\tilde{\mathbf{x}}$ is a good approximation to the ridge regression problem. Setting k properly requires some knowledge of an upper bound $\tilde{s}_\lambda$ on s_λ. A constant factor approximation to s_λ can be computed in $\tilde{O}(n^{3/2} \cdot \operatorname{poly}(s_\lambda))$ time using, for example a column PCP as given by Lemma 2.2.12 and binary searching for an appropriate k value.

An interesting open question is if s_λ be be approximated more quickly – specifically with linear dependence on n. This question is closely related to if it is possible to estimate the cost $\|\mathbf{A} - \mathbf{A}_k\|_F^2$ in $\tilde{O}(n \cdot \operatorname{poly}(k))$ for PSD $\mathbf{A}$, which surprisingly is also open.

2.7 Query Lower Bound

We now present our lower bound on the number of accesses to $\mathbf{A}$ required to compute a near-optimal low-rank approximation, matching the query complexity of Algorithm 2.5.1 up to a $\tilde{O}(1/\epsilon^{1.5})$ factor.

Theorem 2.7.1. *Assume that k, ϵ are such that $nk/\epsilon = o(n^2)$. Consider any (possibly randomized) algorithm $\mathcal{A}$ that, given any PSD $\mathbf{A} \in \mathbb{R}^{n\times n}$, outputs a $(1+\epsilon)$-approximate rank-k approximation to $\mathbf{A}$ in the Frobenius norm (Problem 2.1.1) with probability at least $2/3$. Then there must be some input $\mathbf{A}$ of which $\mathcal{A}$ reads at least $\Omega(nk/\epsilon)$ positions in expectation (over $\mathcal{A}$'s random coin flips).*

2.7.1 Lower Bound Approach

The idea behind Theorem 2.7.1 is to apply Yao's minimax principle [Yao77], proving a lower bound for randomized algorithms on worst-case inputs via a lower bound for deterministic algorithms on some input distribution (a 'hard input distribution'). Specifically, we will draw the input $\mathbf{A}$ from a distribution over binary matrices. $\mathbf{A}$ has all 1's on its diagonal, along with k randomly positioned (non-contiguous) blocks of all 1's, each of size $\sqrt{2\epsilon n/k} \times \sqrt{2\epsilon n/k}$. In other words, $\mathbf{A}$ is the adjacency matrix (plus identity) of a graph with k cliques of size $\sqrt{2\epsilon n/k}$, placed on random subsets of the vertices, with all other vertices isolated.

It is easy to see that every $\mathbf{A}$ chosen from the above distribution is PSD since applying a permutation yields a block diagonal matrix, each of whose blocks is a PSD matrix (either a single 1 entry or a rank-1 all 1's block). Additionally, for every $\mathbf{A}$ chosen from the distribution, the optimal rank-k approximation to $\mathbf{A}$ projects off each of the k blocks, achieving Frobenius norm error $\|\mathbf{A} - \mathbf{A}_k\|_F^2 = n - k\sqrt{2\epsilon n/k} \approx n$. To match this up to a $1+\epsilon$ factor, any near-optimal rank-k approximation must at least capture a constant fraction of the Frobenius norm mass in the blocks since this mass is $k \cdot 2\epsilon n/k = 2\epsilon n$.

Doing so requires identifying at least a constant fraction of the blocks. However, since block positions are chosen uniformly at random, and since the diagonal entries of $\mathbf{A}$ are identical and so convey no information about the positions, to identify a single block, any algorithm essentially must read arbitrary off-diagonal entries until it finds a 1. There are $\approx n^2$ off-diagonal entries with just $2\epsilon n$ of them 1's, so identifying a first block requires $\Omega(n/\epsilon)$ queries to $\mathbf{A}$ in expectation (over the random choice of $\mathbf{A}$). Since the vast majority of vertices are isolated and not contained within a block, finding this first block does little to make finding future blocks easier. So overall, the algorithm must make $\Omega(nk/\epsilon)$ queries in expectation to find a constant fraction of the k blocks and output a near-optimal low-rank approximation with good probability.

While the above intuition is the key idea behind the lower bound, a rigorous proof requires a number of additional steps and modifications, detailed in the remainder of this section. In Section 2.7.2 we introduce the notion of a *primitive approximation* to a matrix, which we will employ in our lower bound for low-rank approximation. In Section 2.7.3 we show that any deterministic low-rank approximation algorithm that succeeds with good probability on the input distribution described above can be used to give an algorithm that computes a primitive approximation with good probability on a matrix drawn from a related input distribution (Lemma 2.7.8). This reduction yields a lower bound for deterministic low-rank approximation algorithms (Theorem 2.7.11), which gives Theorem 2.7.1 after an application of Yao's principle.

2.7.2 Primitive Approximation

We first define the notion of an *ϵ-primitive approximation* to a matrix and establish some basic properties of these approximations.

Definition 2.7.2 (ϵ-primitive Approximation)**.** *A matrix $\mathbf{A}' \in \mathbb{R}^{m \times m}$ is said to be ϵ-primitive for $\mathbf{A} \in \mathbb{R}^{m \times m}$ if the squared Frobenius norm of $\mathbf{A} - \mathbf{A}'$, restricted to its off-diagonal entries is $\leq \epsilon m$. Note that $\mathbf{A}'$ is allowed to have any rank.*

We also define a distribution on matrices with a randomly placed block of all one entries. We will later use this distribution to construct a 'hard input distribution' for deterministic low-rank approximation algorithms.

Definition 2.7.3 (Random Block Matrix)**.** *For any m, ϵ with $m/\epsilon \leq m^2$, let $\mu(m, \epsilon)$ be the distribution on $\mathbf{A} \in \mathbb{R}^{m \times m}$ defined as follows. We choose a uniformly random subset S of $[m]$ where $|S| = \sqrt{16\epsilon m}$, where we assume for simplicity that $|S|$ is an integer. We generate a random matrix $\mathbf{A}$ by setting for each $i \neq j \in S$, $\mathbf{A}_{i,j} = 1$. We then set $\mathbf{A}_{i,i} = 1$ for all i and set all remaining entries of $\mathbf{A}$ to equal 0.*

Note that we associate a random subset S with the sampling of a matrix $\mathbf{A}$ according to $\mu(m, \epsilon)$. It is clear that any $\mathbf{A}$ in the support of $\mu(m, \epsilon)$ is PSD. This is since any $\mathbf{A}$ in the support of $\mu(m, \epsilon)$, after a permutation, is composed of an $|S| \times |S|$ all ones block and an $(m - |S|) \times (m - |S|)$ identity.

If $\mathbf{A}'$ is *ϵ-primitive* for $\mathbf{A}$ in the support of $\mu(m, \epsilon)$, it approximates $\mathbf{A}$ up to error ϵm on its off-diagonal entries. Let R denote the set of off-diagonal entries of $\mathbf{A}$

restricted to the intersection of the rows and columns indexed by S. The error on the entries of R is at most ϵm. Further, restricted to these entries, $\mathbf{A}$ is an all ones matrix. Thus, on the entries of R, both $\mathbf{A}$ and $\mathbf{A}'$ look far from an identity matrix. Formally, we show:

Lemma 2.7.4. *If $\mathbf{A}' \in \mathbb{R}^{m \times m}$ is ϵ-primitive for an $\mathbf{A} \in \mathbb{R}^{m \times m}$ in the support of $\mu(m, \epsilon)$, then $\mathbf{A}'$ is not ϵ-primitive for $\mathbf{I}$.*

Proof. By definition, any ϵ-primitive matrix $\mathbf{A}'$ for $\mathbf{A}$ has squared Frobenius norm restricted to $\mathbf{A}$'s off-diagonal entries of $\leq \epsilon m$. Let R denote the set of off-diagonal entries in the intersection of the rows and columns of $\mathbf{A}$ indexed by S. Assuming $|S| \geq 4$, which follows from the assumption in Definition 2.7.3 that $m/\epsilon \leq m^2$, $|R| = |S|^2 - S \geq 12\epsilon m$.

Restricted to the entries in R, $\mathbf{A}$ is an all ones matrix. Thus, on at least $8\epsilon m$ of these entries $\mathbf{A}'$ must have value at least $1/2$. Otherwise it would have greater than $|R| - 4\epsilon m \geq 4\epsilon m$ entries with value $\leq 1/2$ and so squared Frobenius norm error on $\mathbf{A}$'s off-diagonal entries greater than $\frac{1}{2^2} \cdot 4\epsilon m \geq \epsilon m$, contradicting the assumption that it is ϵ-primitive for $\mathbf{A}$.

Restricted to the entries in R, the identity matrix is all zero. Thus, $\mathbf{A}'$ has Frobenius norm error on these entries at least $\frac{1}{2^2} \cdot 8\epsilon m \geq 2\epsilon m$. Thus $\mathbf{A}'$ is not ϵ-primitive for $\mathbf{I}$, giving the lemma. □

Consider $\mathbf{A}$ drawn from $\mu(m, \epsilon)$ with probability $1/2$ and set to $\mathbf{I}$ with probability $1/2$. By Lemma 2.7.4, any (possibly randomized) algorithm $\mathcal{A}$ that returns an ϵ-primitive approximation to $\mathbf{A}$ with good probability can be used to distinguish with good probability whether $\mathbf{A}$ is in the support of $\mu(m, \epsilon)$ or $\mathbf{A} = \mathbf{I}$. This is because, by Lemma 2.7.4, the output of such an algorithm can only be correct either for some $\mathbf{A}$ in the support of $\mu(m, \epsilon)$ or for the identity. We first define the distribution over matrices to which this result applies:

Definition 2.7.5. *For any m, ϵ with $m/\epsilon \leq m^2$, let $\gamma(m, \epsilon)$ be the distribution on $\mathbf{A} \in \mathbb{R}^{m \times m}$ defined as follows. With probability $1/2$ draw $\mathbf{A}$ from $\mu(m, \epsilon)$ (Definition 2.7.3). Otherwise, draw $\mathbf{A}$ from $\nu(m)$, which is the distribution whose support is only the $m \times m$ identity matrix.*

Note that, as with $\mu(m, \epsilon)$, we associate a random subset S with $|S| = \sqrt{16\epsilon m}$ with the sampling of a matrix $\mathbf{A}$ according to $\gamma(m, \epsilon)$. S is not used in the construction of $\mathbf{A}$ in the case that it is drawn from $\nu(m)$.

Formally, since $\mathcal{A}$ can distinguish whether $\mathbf{A}$ is in the support of $\mu(m, \epsilon)$ or $\nu(m)$ (i.e., $\mathbf{A} = \mathbf{I}$) with good probability we can prove that the distribution of $\mathcal{A}$'s access pattern (over randomness in the input and possible randomization in the algorithm) is significantly different when it is given input $\mathbf{A}$ drawn from $\mu(m, \epsilon)$ than when it is given $\mathbf{A}$ drawn from $\nu(m)$. Recall for distributions α and β supported on elements s of a finite set S, that the total variation distance $D_{TV}(\alpha, \beta) = \sum_{s \in S} |\alpha(s) - \beta(s)|$, where $\alpha(s)$ is the probability of s in distribution α.

Corollary 2.7.6. *Suppose that a (possibly randomized) algorithm $\mathcal{A}$, with probability at least $7/12$ over its random coin flips and random input $\mathbf{A} \in \mathbb{R}^{n \times n}$ drawn from $\gamma(m, \epsilon)$, outputs an ϵ-primitive matrix for $\mathbf{A}$. Further, suppose that $\mathcal{A}$ reads at most r positions of $\mathbf{A}$, possibly adaptively.*[11] *Let S be a random variable indicating the list of positions read and their corresponding values.*[12] *Let $L(\mu)$ denote the distribution of S conditioned on $\mathbf{A} \sim \mu(m, \epsilon)$, and let $L(\nu)$ denote the distribution of S conditioned on $\mathbf{A} \sim \nu(m)$.*[13] *Then*

$$D_{TV}(L(\mu), L(\nu)) \geq 1/6.$$

Proof. By Lemma 2.7.4, if algorithm $\mathcal{A}$ succeeds then its output can be used to decide if $\mathbf{A} \sim \mu(m, \epsilon)$ or if $\mathbf{A} \sim \nu(m)$. The success probability of any such algorithm is well-known (see, e.g., Proposition 2.58 of [BY02]) to be at most $1/2 + D_{TV}(L(\mu), L(\nu))/2$. Making this quantity at least $7/12$ and solving for $D_{TV}(L(\mu), L(\nu))$ proves the corollary. □

2.7.3 Lower Bound for Low-Rank Approximation

We now give a reduction, showing that any deterministic relative error k-rank approximation algorithm that succeeds with good probability on the hard input distribution described in Section 2.7.1 can be used to compute an ϵ-primitive approximation to $\mathbf{A}$ that is drawn from $\gamma(n/(2k), \epsilon)$ with good probability. We first formally define this input distribution.

Definition 2.7.7 (Hard Input Distribution – Low-Rank Approximation)**.** *Suppose $2nk/\epsilon \leq n^2$. Let γ_b be the distribution on $\mathbf{A} \in \mathbb{R}^{n \times n}$ determined as follows. We draw a uniformly random subset S of $[n]$ where $|S| = n/2$, where we assume for simplicity*

[11] That is, for any input $\mathbf{A}$, in any random execution, $\mathcal{A}$ reads at most r entries of $\mathbf{A}$.

[12] S is determined by the random input $\mathbf{A}$ and the random choices of $\mathcal{A}$. Since $\mathcal{A}$ reads at most r positions of any input, we always have $|S| \leq r$.

[13] Here and throughout, we let $\mathbf{A} \sim \mu(m, \epsilon)$ denote the event that, when $\mathbf{A}$ is drawn from the distribution $\gamma(m, \epsilon)$ (Definition 2.7.5), which is a mixture of the distributions $\mu(m, \epsilon)$ and $\nu(m)$, that $\mathbf{A}$ is drawn from $\mu(m, \epsilon)$. $\mathbf{A} \sim \nu(m)$ denotes the analogous event for $\nu(m)$.

that $|S|$ is an integer. We further partition S into k subsets $S^1, S^2, ..., S^k$ chosen uniformly at random. For all $\ell \in [k]$, $|S^\ell| = n/(2k)$, which we also assume to be an integer.

Letting $\mathbf{A}^\ell$ denote the entries of $\mathbf{A}$ restricted to the intersection of the rows and columns indexed by S^ℓ, we independently draw each $\mathbf{A}^\ell$ from $\gamma(n/(2k), \epsilon)$ (Definition 2.7.5).[14] *We then set $\mathbf{A}_{i,i} = 1$ for all i and set all remaining entries of $\mathbf{A}$ to equal 0.*

Our reduction from ϵ-primitive approximation to low-rank approximation is as follows:

Lemma 2.7.8 (Reduction from Primitive Approximation to Low-Rank Approximation)**.** *Suppose that $nk/\epsilon = o(n^2)$ and that $\mathcal{A}$ is a deterministic algorithm that, with probability $\geq 2/3$ on random input $\mathbf{A} \in \mathbb{R}^{n \times n}$ drawn from γ_b, outputs a $(1 + \epsilon/26)$-approximate rank-k approximation to $\mathbf{A}$. Further suppose $\mathcal{A}$ reads at most r positions of $\mathbf{A}$, possibly adaptively.*

Then there is a randomized algorithm $\mathcal{B}$ that, with probability $\geq 7/12$ over its random coin flips and random input $\mathbf{B}$ drawn from $\gamma(n/(2k), \epsilon)$, outputs an ϵ-primitive matrix for $\mathbf{B}$. Further, letting $L(\mu), L(\nu)$ be as defined in Corollary 2.7.6 for $\mathcal{B}$,

$$D_{TV}(L(\mu), L(\nu)) \leq \frac{2\epsilon r}{nk}.$$

Proof. Consider a randomized algorithm $\mathcal{B}$ that, given $\mathbf{B}$ drawn from $\gamma(n/(2k), \epsilon)$ generates a random matrix $\mathbf{A}^{n \times n}$ drawn from γ_b as follows. Choose a uniformly random subset S of $[n]$ with $|S| = n/2$. Partition S into k subsets $S^1, S^2, ..., S^k$ chosen uniformly at random. Note that for $\ell \in [k]$ $|S^\ell| = n/(2k)$. Letting $\mathbf{A}^\ell$ denote the entries of $\mathbf{A}$ restricted to the intersection of the rows and columns indexed by S^ℓ, set $\mathbf{A}^1 = \mathbf{B}$, and for $\ell = 2, ..., k$ independently draw $\mathbf{A}^\ell$ from $\gamma(n/(2k), \epsilon)$. Set $\mathbf{A}_{i,i} = 1$ for all i and set all remaining entries of $\mathbf{A}$ equal to 0.

After generating $\mathbf{A}$, $\mathcal{B}$ then applies $\mathcal{A}$ to $\mathbf{A}$ to compute a rank-k approximation $\mathbf{A}'$. $\mathcal{B}$ then outputs $(\mathbf{A}')^1$, the $n/(2k) \times n/(2k)$ submatrix of $\mathbf{A}'$ corresponding to the intersection of the rows and columns indexed by S^1. We have the following:

Claim 2.7.9. *With probability $\geq 7/12$ over the random choices of $\mathcal{B}$ and over the random input $\mathbf{B}$ drawn from $\gamma(n/(2k), \epsilon)$, $(\mathbf{A}')^1$ is ϵ-primitive for $\mathbf{B}$.*

Proof. It is clear that $\mathbf{A}$ generated by $\mathcal{B}$ is distributed according to γ_b (Definition 2.7.7). By construction, for any $\mathbf{A} \sim \gamma_b$ there is a rank-k approximation of cost

[14] By the assumption that $2nk/\epsilon \leq n^2$ we have $\frac{n}{2k\epsilon} \leq \left(\frac{n}{2k}\right)^2$ and so this is a valid setting of the parameters for $\gamma(m, \epsilon)$.

at most n; indeed this follows by choosing the best rank-1 solution for each $\mathbf{A}^\ell$. Consequently if $\mathcal{A}$ succeeds on $\mathbf{A}$, then its output $\mathbf{A}'$ satisfies $\|\mathbf{A}-\mathbf{A}'\|_F^2 \leq n+\epsilon n/26$.

Note that $\mathbf{A}$ has $\mathbf{A}_{i,i} = 1$ for all $i \in [n]$. So the squared Frobenius norm cost of any rank-k approximation restricted to the diagonal is at least $n-k$. Let c_ℓ be the squared Frobenius norm cost of $\mathbf{A}'$ restricted to the off diagonal entries in $\mathbf{A}^\ell$. Note that c_ℓ is a random variable. Then,

$$n-k+\sum_{\ell=1}^{k} c_\ell \leq \|\mathbf{A}'-\mathbf{A}\|_F^2 \leq n+\epsilon n/26.$$

By averaging for at least a 11/12 fraction of the blocks i,

$$c_i \leq 12+12\epsilon n/(26k) \leq 13\epsilon n/(26k) < \epsilon n/(2k),$$

assuming $\epsilon n/(26k) \geq 8$, which holds if $\epsilon n/k = \omega(1)$, which follows from our assumption that $nk/\epsilon = o(n^2)$.

It follows by symmetry of γ_b with respect to the k blocks that with probability at least 11/12, if $\mathcal{A}$ succeeds on $\mathbf{A}$, $c_1 \leq \epsilon n/(2k)$. This gives that $(\mathbf{A}')^1$ is ϵ-primitive (Definition 2.7.2) for $\mathbf{A}^1 = \mathbf{B}$. This yields the claim after applying a union bound, since $\mathcal{A}$ succeeds with probability at least 2/3 on $\mathbf{A}$. □

It remains to bound the total variation distance between $\mathcal{B}$'s access pattern when $\mathbf{B} \sim \mu(n/(2k),\epsilon)$ and when $\mathbf{B} \sim \nu(n/(2k))$. We have:

Claim 2.7.10. *For the algorithm $\mathcal{B}$ defined above*

$$D_{TV}(L(\mu), L(\nu)) \leq \frac{2\epsilon r}{nk}.$$

Proof. Let $\mathcal{W}$ denote the random variable that encompasses $\mathcal{B}$'s random choices in choosing the indices in $S^2,...,S^k$ and in setting the entries in $\mathbf{A}^2,...,\mathbf{A}^k$. Let Ω denote the set of all possible values of $\mathcal{W}$. Let S denote the random subset with $|S| = \sqrt{n/(2k)\cdot\epsilon}$ associated with the drawing of $\mathbf{B}$ from $\gamma(n/(2k),\epsilon)$ (see Definition 2.7.5). Let $\bar{S}$ denote the set of off-diagonal entries of $\mathbf{A}$ in the intersection of the rows and columns indexed by S.

If $\mathbf{B} \sim \mu(n/(2k),\epsilon)$ then all entries of $\bar{S}$ are 1. If $\mathbf{B} \sim \nu(n/(2k))$ then $\mathbf{B} = \mathbf{I}$ and so all entries of $\bar{S}$ are 0. Outside of the entries in $\bar{S}$, the entries of $\mathbf{B}$ are identical in the two cases that $\mathbf{B} \sim \mu(n/(2k),\epsilon)$ and $\mathbf{B} \sim \nu(n/(2k))$ (in particular, they are all 0 off the diagonal and 1 on the diagonal).

So, for any $w \in \Omega$, conditioned on $\mathcal{W} = w$, all entries outside $\bar{S}$ are fixed. Further conditioned on $\mathbf{B} \sim \nu(n/(2k))$, all entries in $\bar{S}$ are 0. So all entries of $\mathbf{A}$ are fixed and $\mathcal{A}$ always reads the same sequence of entries $(i_{w,1}, j_{w,1}), ..., (i_{w,r}, j_{w,r})$.

Further, for any $w \in \Omega$, conditioned on $\mathcal{W} = w$, S is a uniform random subset of $R = [n] \setminus (S^2 \cup ... \cup S^k)$. $|R| = n - (k-1) \cdot n/(2k) \geq n/2$. So, for any $\ell \in [r]$ we have:

$$\mathbb{P}[(i_{w,\ell}, j_{w,\ell}) \in \bar{S} | \mathcal{W} = w] \leq \frac{|\bar{S}|}{|R|^2} \leq \frac{|S|^2}{n^2/4} \leq \frac{2\epsilon}{nk}.$$

By a union bound, letting $\mathcal{E}$ be the event that $\mathcal{A}$ reads $(i_{w,1}, j_{w,1}), ..., (i_{w,r}, j_{w,r})$ and does not read any entry of $\bar{S}$ in its r accesses to $\mathbf{A}$ we have for any $w \in \Omega$:

$$\mathbb{P}[\mathcal{E} | \mathcal{W} = w] \geq 1 - \frac{2\epsilon r}{nk}.$$

Thus, for any $w \in \Omega$, regardless of whether $\mathbf{B} \sim \nu(n/(2k))$ or $\mathbf{B} \sim \mu(n/(2k), \epsilon)$, $\mathcal{A}$ has access pattern $(i_{w,1}, j_{w,1}), ..., (i_{w,r}, j_{w,r})$ to $\mathbf{A}$ with probability $\geq 1 - \frac{2\epsilon r}{nk}$. Correspondingly, $\mathcal{B}$ has a fixed access pattern to $\mathbf{B}$ with probability $\geq 1 - \frac{2\epsilon r}{nk}$. Thus,

$$D_{TV}(L(\mu), L(\nu)) \leq \frac{2\epsilon r}{nk}$$

yielding the claim. □

In combination Claims 2.7.9 and 2.7.10 give Lemma 2.7.8. □

We can now use Lemma 2.7.8 to argue that if a deterministic low-rank approximation algorithm succeeding with good probability on a random input drawn from γ_b accesses too few entries, then it can be used to give a primitive approximation algorithm violating Corollary 2.7.6. We then prove our main lower bound, Theorem 2.7.1, by applying Yao's minimax principle.

Theorem 2.7.11 (Lower Bound for Deterministic Algorithms)**.** *Assume that n, k, ϵ are such that $nk/\epsilon = o(n^2)$. Consider any deterministic algorithm $\mathcal{A}$ that, given random input $\mathbf{A}$ drawn from γ_b, outputs a $(1+\epsilon)$-approximate rank-k approximation to $\mathbf{A}$ in the Frobenius norm (Problem 2.1.1) with probability at least $2/3$. Further, suppose $\mathcal{A}$ reads at most r positions of $\mathbf{A}$, possibly adaptively. Then $r = \Omega(nk/\epsilon)$.*

Proof. Assume towards a contradiction that $r = o(nk/\epsilon)$. Then applying Lemma 2.7.8, $\mathcal{A}$ can be used to give a randomized algorithm $\mathcal{B}$ that with probability $\geq 7/12$ over its random coin flips and random input $\mathbf{B} \in n/(2k) \times n/(2k)$ drawn from $\gamma(n/(2k), \epsilon)$, outputs an ϵ-primitive matrix for $\mathbf{B}$. Further, letting $L(\mu), L(\nu)$ be as defined in Corollary 2.7.6 for $\mathcal{B}$, by Lemma 2.7.8, $D_{TV}(L(\mu), L(\nu)) \leq \frac{2\epsilon r}{nk}$. For $r = o(nk/\epsilon)$, $\frac{2\epsilon r}{nk} = o(1)$, contradicting Corollary 2.7.6, and giving the theorem. □

We finally restate and prove our main lower bound.

Theorem 2.7.1 (Lower Bound for Randomized Algorithms)**.** *Assume that n, k, ϵ are such that $nk/\epsilon = o(n^2)$. Consider any (possibly randomized) algorithm $\mathcal{A}$ that, given any PSD $\mathbf{A} \in \mathbb{R}^{n\times n}$, outputs a $(1+\epsilon)$-approximate rank-k approximation to $\mathbf{A}$ in the Frobenius norm (Problem 2.1.1) with probability at least $2/3$. Then there must be some input $\mathbf{A}$ on which $\mathcal{A}$ reads $\Omega(nk/\epsilon)$ positions in expectation (over $\mathcal{A}$'s random coin flips).*

Proof. This follows directly from applying Yao's minimax principle ([Yao77], Theorem 3) to Theorem 2.7.11. □

2.8 Low-Rank Approximation of A via $\mathbf{A}^{1/2}$

In this section we discuss computing a low-rank approximation of PSD $\mathbf{A}$ using a low-rank approximation for the matrix square root $\mathbf{A}^{1/2}$. We demonstrate that, while naively, a near-optimal low-rank approximation for $\mathbf{A}^{1/2}$ can be arbitrarily bad for $\mathbf{A}$, there is a way to convert a low-rank approximation from $\mathbf{A}^{1/2}$ to one for $\mathbf{A}$. This method gives a simple, but suboptimal, sublinear time low-rank approximation algorithm for PSD matrices. The purpose of this section is to build intuition for PSD low-rank approximation and connections between a PSD matrix $\mathbf{A}$ and its squareroot $\mathbf{A}^{1/2}$. We leveraged these connections in a different way in designing our main low-rank approximation algorithm, Algorithm 1, presented in Section 2.5.

2.8.1 Converting a Low-Rank Approximation of $\mathbf{A}^{1/2}$ to a Low-Rank Approximation of A

We first observe that a low-rank approximation for $\mathbf{A}^{1/2}$ does not imply a good low-rank approximation for $\mathbf{A}$. Intuitively, if $\mathbf{A}$ has a large top singular value, the low-rank approximation for $\mathbf{A}$ must capture the corresponding singular direction with significantly more accuracy than a good low-rank approximation for $\mathbf{A}^{1/2}$, in which the singular value is relatively much smaller.

Theorem 2.8.1. *For any k, $\epsilon > 0$ there exists a PSD matrix $\mathbf{A}$ and a rank-k matrix $\mathbf{B}$ such that $\|\mathbf{A}^{1/2} - \mathbf{B}\|_F^2 \leq (1+\epsilon)\|\mathbf{A}^{1/2} - (\mathbf{A}^{1/2})_k\|_F^2$ but for every matrix $\mathbf{C}$ in the* *row span of* $\mathbf{B}$*,*

$$\|\mathbf{A}-\mathbf{C}\|_F^2 \geq \left(1+\epsilon\cdot\frac{(n-k-1)\sigma_1(\mathbf{A})}{\sigma_{k+1}(\mathbf{A})}\right)\|\mathbf{A}-\mathbf{A}_k\|_F^2.$$

Notably, if we $\mathbf{B}=\mathbf{A}^{1/2}\mathbf{P}$ for some rank-k orthogonal projection $\mathbf{P}$, $\mathbf{AP}$ can be an arbitrarily bad low-rank approximation of $\mathbf{A}$.

Proof. Let $\mathbf{A}\in\mathbb{R}^{n\times n}$ be a diagonal matrix with $\mathbf{A}_{i,i}=\alpha^2$ for $i\in[k]$, $\mathbf{A}_{k+1,k+1}=0$ and all other diagonal entries equal to β^2, where $\alpha>\beta>0$. Let $\mathbf{B}$ be a rank-k matrix that has its last $n-k$ rows all zero. For $i\in[k]$, let $\mathbf{B}_{i,i}=\mathbf{A}_{i,i}^{1/2}$ and $\mathbf{B}_{1,k+2}=\sqrt{\epsilon(n-k-1)}\cdot\beta$. We have: $\|\mathbf{A}^{1/2}-(\mathbf{A}^{1/2})_k\|_F^2=(n-k-1)\beta^2$ and $\|\mathbf{A}^{1/2}-\mathbf{B}\|_F^2=(1+\epsilon)(n-k-1)\beta^2$. Note that the first row of $\mathbf{B}$, $\mathbf{b}_1$ aligns somewhat well with the first row of $\mathbf{A}$, $\mathbf{a}_1$, but as we will see, not well enough to give a good low-rank approximation for $\mathbf{A}$ itself.

Let $\mathbf{C}$ be the projection of $\mathbf{A}$ onto the row span of $\mathbf{B}$, which gives the optimal low-rank approximation to $\mathbf{A}$ within this span. For $i=2,...,k$, $\mathbf{c}_i=\mathbf{a}_i$, since $\mathbf{A}$ and $\mathbf{B}$ match exactly on these rows up to a scaling. For $i>k$, $\mathbf{c}_i=\vec{0}$. Finally, $\mathbf{c}_1=\frac{\mathbf{b}_1}{\|\mathbf{b}_1\|_2^2}\cdot\langle\mathbf{b}_1,\mathbf{a}_1\rangle=\mathbf{b}_1\cdot\left(\frac{\alpha^3}{\alpha^2+\epsilon(n-k-1)\beta^2}\right)$. Overall:

$$\begin{aligned}\|\mathbf{A}-\mathbf{C}\|_F^2 &= (n-k-1)\beta^4+(\mathbf{A}_{1,1}-\mathbf{C}_{1,1})^2+(\mathbf{A}_{1,k+2}-\mathbf{C}_{1,k+2})^2\\ &\geq (n-k-1)\beta^4+\left(\frac{\sqrt{\epsilon(n-k-1)}\cdot\beta\alpha^3}{\alpha^2+\epsilon(n-k-1)\beta^2}\right)^2\\ &\geq (n-k-1)\beta^4\cdot(1+\epsilon(n-k-1)\alpha^2/4\beta^2)\\ &= (1+\epsilon(n-k-1)\alpha^2/\beta^2)\cdot\|\mathbf{A}-\mathbf{A}_k\|_F^2.\end{aligned}$$

By setting $\alpha\gg\beta$ we can make this approximation arbitrarily bad. Note that $\alpha^2/\beta^2=\sigma_1(\mathbf{A})/\sigma_{k+1}(\mathbf{A})$. This ratio will be large whenever $\mathbf{A}$ is well approximated by a low-rank matrix. □

Despite the above example, we can show that for a projection $\mathbf{P}$, if $\mathbf{A}^{1/2}\mathbf{P}$ is a very near-optimal low-rank approximation of $\mathbf{A}^{1/2}$ then $\mathbf{A}^{1/2}\mathbf{P}\mathbf{A}^{1/2}$ is a relative error low-rank approximation of $\mathbf{A}$:

Theorem 2.8.2. *Let* $\mathbf{P}\in\mathbb{R}^{n\times n}$ *be an orthogonal projection matrix such that* $\|\mathbf{A}^{1/2}-\mathbf{A}^{1/2}\mathbf{P}\|_F^2\leq(1+\epsilon/\sqrt{n})\|\mathbf{A}^{1/2}-(\mathbf{A}^{1/2})_k\|_F^2$*. Then:*

$$\|\mathbf{A}-\mathbf{A}^{1/2}\mathbf{P}\mathbf{A}^{1/2}\|_F^2\leq(1+3\epsilon)\|\mathbf{A}-\mathbf{A}_k\|_F^2.$$

Proof. We can rewrite using the fact that $(\mathbf{I}-\mathbf{P}) = (\mathbf{I}-\mathbf{P})^2$ since it is a projection:

$$\|\mathbf{A} - \mathbf{A}^{1/2}\mathbf{P}\mathbf{A}^{1/2}\|_F^2 = \|\mathbf{A}^{1/2}(\mathbf{I}-\mathbf{P})^2\mathbf{A}^{1/2}\|_F^2 = \|\mathbf{A}^{1/2}(\mathbf{I}-\mathbf{P})\|_4^4.$$

Let $\delta_i = \sigma_i(\mathbf{A}^{1/2}(\mathbf{I}-\mathbf{P}))$ denote the i^{th} singular value of $\mathbf{A}^{1/2}(\mathbf{I}-\mathbf{P})$. By the assumption that $\mathbf{P}$ gives a near-optimal low-rank approximation of $\mathbf{A}^{1/2}$:

$$\sum_{i=1}^{n-k} \delta_i^2 \leq \sum_{i=k+1}^{n} \sigma_i(\mathbf{A}) + \epsilon/\sqrt{n}\|\mathbf{A}^{1/2} - (\mathbf{A}^{1/2})_k\|_F^2.$$

Additionally, by Weyl's monotonicity theorem (see e.g. Theorem 3.2 in [Gu14] and proof of Lemma 15 in [MM17]), for all i, $\delta_i \geq \sigma_{i+k}^{1/2}(\mathbf{A})$. We thus have:

$$\|\mathbf{A} - \mathbf{A}^{1/2}\mathbf{P}\mathbf{A}^{1/2}\|_F^2 = \sum_{i=1}^{n-k} \delta_i^4 \leq \sum_{i=k+2}^{n} \sigma_i(\mathbf{A})^2 + \left(\sigma_{k+1}(\mathbf{A}) + \epsilon/\sqrt{n}\|\mathbf{A}^{1/2} - (\mathbf{A}^{1/2})_k\|_F^2\right)^2.$$

If $\sigma_{k+1}(\mathbf{A}) \geq 1/\sqrt{n} \cdot \|\mathbf{A}^{1/2} - (\mathbf{A}^{1/2})_k\|_F^2$ then

$$\begin{aligned}\left(\sigma_{k+1}(\mathbf{A}) + \epsilon/\sqrt{n}\|\mathbf{A}^{1/2} - (\mathbf{A}^{1/2})_k\|_F^2\right)^2 &\leq (1+\epsilon)^2\sigma_{k+1}^2(\mathbf{A}) \\ &\leq (1+3\epsilon)\sigma_{k+1}^2(\mathbf{A})\end{aligned}$$

and hence:

$$\|\mathbf{A} - \mathbf{A}^{1/2}\mathbf{P}\mathbf{A}^{1/2}\|_F^2 \leq (1+3\epsilon)\sum_{i=k+1}^{n} \sigma_i^2(\mathbf{A}) = (1+3\epsilon)\|\mathbf{A} - \mathbf{A}_k\|_F^2.$$

Alternatively if $\sigma_{k+1}(\mathbf{A}) \leq 1/\sqrt{n} \cdot \|\mathbf{A}^{1/2} - (\mathbf{A}^{1/2})_k\|_F^2$ then

$$\begin{aligned}\left(\sigma_{k+1}(\mathbf{A}) + \epsilon/\sqrt{n}\|\mathbf{A}^{1/2} - (\mathbf{A}^{1/2})_k\|_F^2\right)^2 &\leq \left((1+\epsilon)/\sqrt{n}\|\mathbf{A}^{1/2} - (\mathbf{A}^{1/2})_k\|_F^2\right)^2 \\ &\leq (1+\epsilon)^2\|\mathbf{A} - \mathbf{A}_k\|_F^2,\end{aligned}$$

which also gives the theorem. □

2.8.2 PSD Low-Rank Approximation in $n^{1.69} \cdot \mathrm{poly}(k/\epsilon)$ Time

We now combine Theorem 2.8.2 with the ridge leverage score sampling algorithm of Lemma 2.2.17 to give a sublinear time algorithm for low-rank approximation of $\mathbf{A}$ reading $n^{3/2} \cdot \mathrm{poly}(k/\epsilon)$ entries of the matrix and running in $n^{1.69} \cdot \mathrm{poly}(k/\epsilon)$ time. We note that this approach could also be used with adaptive sampling [DV06] or volume sampling techniques [AGR16], as outlined in the introduction.

Theorem 2.8.3. *There is an algorithm based off ridge leverage score sampling that, given PSD* $\mathbf{A} \in \mathbb{R}^{n \times n}$*,* $k \in \mathbb{Z}^{\geq 1}$*, and* $\epsilon, \delta \in (0, 1/2]$*, outputs* $\mathbf{M} \in \mathbb{R}^{n \times k}$ *such that, with probability* $\geq 1 - \delta$ *,*

$$\|\mathbf{A} - \mathbf{M}\mathbf{M}^T\|_F^2 \leq (1+\epsilon)\|\mathbf{A} - \mathbf{A}_k\|_F^2.$$

The algorithm reads $\tilde{O}(n^{3/2}k/\epsilon)$ *entries of* $\mathbf{A}$ *and runs in* $\tilde{O}\left(n^{(\bar{\omega}+1)/2} \cdot (k/\epsilon)^{\bar{\omega}-1}\right)$ *time, where* $\tilde{O}(\cdot)$ *hides* $\log(n/\delta)$ *dependencies.*

Note that for $\bar{\omega} < 2.373$ [LG14], $n^{(\bar{\omega}+1)/2} \leq n^{1.69}$. Theorem 2.8.3 follows from Lemma 2.2.17, adapted from Theorem 8 of [MM17], which shows that it is possible to estimate the ridge leverage scores of $\mathbf{A}^{1/2}$ with $\tilde{O}(nk)$ accesses to $\mathbf{A}$ and $O(nk^{\bar{\omega}-1})$ time. We can use these scores to sample a set of rows from $\mathbf{A}^{1/2}$ whose span contains a near-optimal low-rank approximation. Specifically we have:

Lemma 2.8.4 (Theorem 7 of [CMM17])**.** *For any* $\mathbf{B} \in \mathbb{R}^{n \times n}$*, for every* $i \in [n]$*, let* $\tilde{\tau}_i^k \geq \tau_i^k(\mathbf{B})$ *be an overestimate for the* i^{th} *rank-*k *ridge leverage score of* $\mathbf{B}$*. Let* $p_i = \frac{\tilde{\tau}_i^k}{\sum_i \tilde{\tau}_i^k}$ *and for* $\epsilon, \delta \in (0, 1/2]$*,* $t = c\left(\log k + \frac{\log(1/\delta)}{\epsilon}\right)\sum_i^k \tilde{\tau}_i^k$ *for some sufficiently large constant* c*. Construct* $\mathbf{R} \in \mathbb{R}^{n \times t}$ *by sampling* t *rows of* $\mathbf{B}$*, each set to row* $\mathbf{b}_i$ *with probability* p_i*. With probability* $\geq 1 - \delta$*, letting* $\mathbf{P}_\mathbf{R}$ *be the projection onto the rows of* $\mathbf{R}$*,*

$$\|\mathbf{B} - (\mathbf{B}\mathbf{P}_\mathbf{R})_k\|_F^2 \leq (1+\epsilon)\|\mathbf{B} - \mathbf{B}_k\|_F^2.$$

Note that $(\mathbf{B}\mathbf{P}_\mathbf{R})_k$ can be written as a row projection of $\mathbf{B}$ – onto the top k singular vectors of $\mathbf{B}\mathbf{P}_\mathbf{R}$.

Proof of Theorem 2.8.3. If we compute for each i, $\tilde{\tau}_i^k \geq \tau_i^k(\mathbf{A}^{1/2})$ using Lemma 2.2.17, set $\epsilon' = \epsilon/3\sqrt{n}$, and let $\mathbf{S}$ be a sampling matrix selecting $\tilde{O}(\sum \tilde{\tau}_i^k/\epsilon') = \tilde{O}(k\sqrt{n}/\epsilon)$ rows of $\mathbf{A}^{1/2}$, then by Theorems 2.8.2 and 2.8.4, letting $\mathbf{P}$ be the projection onto the rows of $\mathbf{S}\mathbf{A}^{1/2}$, with probability $\geq 1 - \delta$,

$$\|\mathbf{A} - (\mathbf{A}^{1/2}\mathbf{P})_k(\mathbf{A}^{1/2}\mathbf{P})_k^T\|_F^2 \leq (1+\epsilon)\|\mathbf{A} - \mathbf{A}_k\|_F^2.$$

We can write

$$(\mathbf{A}^{1/2}\mathbf{P})_k(\mathbf{A}^{1/2}\mathbf{P})_k^T = (\mathbf{A}^{1/2}\mathbf{P}\mathbf{A}^{1/2})_k = (\mathbf{A}\mathbf{S}(\mathbf{S}^T\mathbf{A}\mathbf{S})^+\mathbf{S}^T\mathbf{A})_k.$$

We can compute a factorization of this matrix by computing $(\mathbf{S}^T\mathbf{A}\mathbf{S})^{+/2}$, then computing $\mathbf{A}\mathbf{S}(\mathbf{S}^T\mathbf{A}\mathbf{S})^{+/2}$ and taking the SVD of this matrix. Since $\mathbf{S}$ has $\tilde{O}(k\sqrt{n}/\epsilon)$ columns, using fast matrix multiplication this requires time $\tilde{O}(n \cdot (k\sqrt{n}/\epsilon)^{\bar{\omega}-1}) = \tilde{O}(n^{(\bar{\omega}+1)/2} \cdot (k/\epsilon)^{\bar{\omega}-1})$ and $\tilde{O}(n^{3/2}k/\epsilon)$ accesses to $\mathbf{A}$ (to read the entries of $\mathbf{A}\mathbf{S}$), giving the theorem. □

2.9 Discussion and Future Work

In this Chapter, we have presented sublinear time algorithms for low-rank approximation of PSD matrices (Sections 2.5 and 2.6) and ridge regression involving PSD inputs (Section 2.6). These algorithms are based on random leverage score sampling (Sections 2.2, 2.3, and 2.4), which allows us to rapidly reduce any PSD input to a small subset of rows and columns from which we can compute a near-optimal low-rank approximation to the input. We now discuss a number of research directions left open by our work. We note that one direction we are interested in is the possibility of randomized linear algebraic techniques being implemented in neural networks. We defer discussion of this direction until Chapter 5, Section 5.6.

2.9.1 Sublinear Time Algorithms for PSD Matrices

Our work opens the possibility for developing sublinear time algorithms for PSD matrices beyond the foundational low-rank approximation problems we have considered.

One interesting direction is to consider low-rank approximation in norms other than the Frobenius norm, which we considered in Problem 2.1.1. For example, it would be interesting to design algorithms achieving relative error approximation guarantees in general Schatten-p norms and in the entrywise ℓ_1 and ℓ_p norms [SWZ17, CGK+17], which are often used as more 'robust' alternatives to Frobenius norm low-rank approximation. As discussed, obtaining low-rank approximation error guarantees in, for example, higher Schatten-p norms will require increased runtime. Specifically, using a similar argument to the lower bound presented in Section 2.7, one can show that relative error low-rank approximation in $\|\mathbf{A} - \mathbf{B}\|_p^p = \sum_{i=1}^n \sigma_i^p(\mathbf{A} - \mathbf{B})$ for $p > 2$ requires $\Omega(n^{2-2/p})$ queries to $\mathbf{A}$ (see discussion in Section 2.1.4). Understanding if this lower bound can be matched is an interesting direction.

In general, understanding what properties of a PSD matrix can be computed in sublinear time is interesting. For example, while we can output $\mathbf{B}$ satisfying $\|\mathbf{A} - \mathbf{B}\|_F^2 \leq (1+\epsilon)\|\mathbf{A} - \mathbf{A}_k\|_F^2$, surprisingly it is not clear how to actually estimate the value $\|\mathbf{A}-\mathbf{A}_k\|_F^2$ to within a $(1\pm\epsilon)$ factor. This can be achieved in $n^{3/2}\operatorname{poly}(k/\epsilon)$ time using the PCP techniques described in Sections 2.3 and 2.4. However, obtaining linear runtime in n is open. Estimating $\|\mathbf{A}-\mathbf{A}_k\|_F^2$ seems strongly connected to estimating other important quantities such as the statistical dimension of $\mathbf{A}$, $\operatorname{tr}((\mathbf{A}^2+\lambda\mathbf{I})^{-1}\mathbf{A})$, for ridge regression (see Theorem 2.6.7), which we also do not know how to do in $o(n^{3/2})$ time.

Finally, it would be very interesting to understand if sublinear time low-rank approximation algorithms can be achieved deterministically. Even input sparsity runtimes, which are state-of-the-art for general (not necessarily PSD) matrices [CW13], are not known to be achievable with deterministic algorithms. Studying deterministic methods could lead to the development of new techniques for approximate linear algebraic computation, outside of random sampling and sketching, with applications beyond low-rank approximation. Alternatively, it may be possible to prove lower bounds against deterministic algorithms. Such lower bounds information theoretic, like the bound presented in Section 2.7, or may use reduction-based arguments similar to those in Chapter 3.

2.9.2 Sublinear Time Algorithms for Other Matrix Types

It would also be interesting to understand if our techniques can be generalized to a broader class of inputs, beyond PSD matrices. As discussed, in the matrix completion literature, much attention has focused on incoherent low-rank matrices [CR09, PLSZ17], which can be approximated in sublinear time using uniform sampling techniques. PSD matrices are not incoherent in general, which is highlighted by the fact that our sampling schemes are far from uniform and very adaptive to previously seen matrix entries. However, it would be interesting to study other parameters that characterize when low-rank approximation can be performed with just a small number of adaptive accesses to $\mathbf{A}$. For example, we may consider:

- Measures of diagonal dominance that encompass the fact, noted in Section 2.1.1, that for any PSD $\mathbf{A}$ and any i, j, $|\mathbf{A}_{i,j}| \leq \max(\mathbf{A}_{i,i}, \mathbf{A}_{j,j})$.

- Letting $\mathbf{A} = \mathbf{A}^{1/2}\mathbf{A}^{1/2}$, we can think of $\mathbf{A}$ as containing all pairwise dot products between the rows of $\mathbf{A}^{1/2}$. We may consider low-rank approximation of non-PSD matrices who entries contain other pairwise similarity measurements between a set of n-points.

- Matrices that are not PSD, but are close to PSD. For example, matrices that are well approximated by a PSD low-rank approximation, have only a few, bounded negative eigenvalues, or are PSD matrices with a few corrupted entries.

2.9.3 Expanding the Applications of Leverage Scores

Our sublinear time low-rank approximation algorithms critically relied on approximation bounds given by ridge leverage score sampling of matrix rows and columns (see Section 2.2). As discussed, variations on these scores have been employed in algorithms for ordinary least squares regression [DMM06a, LMP13, CLM$^+$15], ℓ_p norm regression [CP15], kernel ridge regression [AM15a, AKM$^+$17], CUR matrix approximation [MD09], graph sparsification [KLM$^+$17], fast system solvers [KLP$^+$16, KS16], second order optimization [ABH16, LHLS17], and many other problems. In all of these applications, leverage score sampling is used due to its ability to approximate a matrix, typically in the sense of the spectral approximation bounds given in Lemma 2.2.6 and Corollary 2.2.11.

An important direction for future work is studying how leverage scores, and related importance measures, can be used to approximate input data in other ways. For example, the leverage scores are a special case of *influence functions* [TB17], which generally measure how important a single data point is on the output of an algorithm. Sampling by general influence functions may allow some of the progress made with leverage score sampling to be extended beyond linear algebraic problems, to other optimization problems like logistic regression or generalized linear models [McC84]. Additionally, while fast approximation algorithms for general influence functions are not known, applying the iterative sampling techniques that have been developed to approximate leverage scores [CLM$^+$15, CMM17, MM17], is a promising possibility.

Chapter 3

Lower Bounds for Linear Algebraic Computation

In Chapter 2 we showed an example of how randomization and approximation can be used to significantly accelerate the basic linear algebraic problem of low-rank approximation. As discussed, in the last decade, progress has been made on fast algorithms for a wide range of basic problems including linear regression (linear system solving), matrix function approximation, eigenvector approximation, and spectrum summarization. In this chapter we attempt to understand the limits of this progress and the importance of randomization and approximation in yielding faster algorithms for linear algebraic computation.

For essentially all nontrivial linear algebraic primitives, there are no known fast, deterministic, and exact algorithms that work for general matrices. Specifically, all existing algorithms for primitives such as linear system solving, eigenvector computation, and determinant computation fall into one of three categories:

1. *They make assumptions on the input matrix* – e.g., that it is well-conditioned (the ratio of its smallest to largest singular value is not too small), has sufficient gaps between its singular values, or is structured (e.g., that it is tridiagonal, triangular, Toeplitz, or symmetric diagonally dominant).

2. *They are randomized and approximate*, typically succeeding with high probability and achieving $(1+\epsilon)$ error guarantees in an appropriate metric with a runtime depending on a low-degree polynomial in $1/\epsilon$.

3. *They are slow*, running in roughly the same time as general matrix multiplication. That is, they require $\Omega(n^{\omega})$ time for $n \times n$ inputs in theory and $\Omega(n^3)$

time in practice, where $\omega < 2.373$ is the greatest lower bound on the exponent of fast matrix multiplication.[1] These algorithms, known as 'direct methods' are generally too slow to apply to large matrices.

As far we are aware, these classes cover all known linear algebraic algorithms. In particular, there are *no fast (i.e., $o(n^\omega)$ time) algorithms that work for general matrices and are not randomized or approximate.*

In this chapter we make a few initial steps in explaining this phenomenon via conditional lower bounds that reduce matrix multiplication to other basic linear algebraic primitives, showing that these primitives are in some sense as hard as matrix multiplication itself. We cover work originally published in [MNS+18] that gives conditional lower bounds for algorithms (both deterministic and randomized) computing many properties of a matrix's singular value spectrum, such as its nuclear norm, its determinant and its SVD entropy, to high accuracy. These lower bounds demonstrate a tradeoff between runtime and approximation accuracy in solving these fundamental problems – showing that any algorithm computing a fine enough approximation essentially must run in matrix multiplication time. We also cover work published in [MW17a] that gives conditional lower bounds on low-rank approximation algorithms for kernel matrices. Aside from conditional lower bounds, in both papers we also give algorithmic results. We do not cover these algorithmic results in detail, but state the main bounds given and discuss how they compare to our lower bounds.

Remark: The results presented in this chapter were developed largely in collaboration with David Woodruff. The lower bounds on effective resistance and leverage score computation presented in Section 3.2.4 were also developed in collaboration with Aaron Sidford. Aaron originally wrote up the details of these bounds; the presentation has been minorly modified in this thesis.

3.1 Background and Introduction to Results

We begin by defining the two main classes of problems we consider in this chapter and giving background on their applications and state-of-the-art runtimes.

3.1.1 Spectral Sum Problems

Given $\mathbf{A} \in \mathbb{R}^{n \times d}$, a central primitive in numerical computation and data analysis is to compute $\mathbf{A}$'s spectrum: the singular values $\sigma_1(\mathbf{A}) \geq \ldots \geq \sigma_d(\mathbf{A}) \geq 0$. These values

[1] See Section 1.3 for a formal definition of ω.

can reveal matrix structure and low effective dimensionality, which can be exploited in a wide range of spectral data analysis methods [Jol02, US16]. The singular values are also used as tuning parameters in many numerical algorithms performed on $\mathbf{A}$ [GVL12], and in general, to determine some of the most well-studied matrix functions [Hig08]. We focus in particular on conditional lower bounds for computing one of the most widely applicable classes of matrix functions that depend on the singular values: *spectral sums.*

Definition 3.1.1 (Spectral Sum)**.** *For any function $f : \mathbb{R}^+ \to \mathbb{R}^+$, and any matrix $\mathbf{A} \in \mathbb{R}^{n\times d}$ define the* spectral sum*:*

$$\mathcal{S}_f(\mathbf{A}) \overset{\text{def}}{=} \sum_{i=1}^{d} f(\sigma_i(\mathbf{A})).$$

Spectral sums often serve as snapshots of $\mathbf{A}$'s spectrum. We list the best known examples:

- The nuclear norm (also called the trace norm), denoted $\|\mathbf{A}\|_1$, is given by $\mathcal{S}_f(\mathbf{A})$ for $f(x) = x$.
- The Schatten p-norm, denoted $\|\mathbf{A}\|_p^p$, is given by $\mathcal{S}_f(\mathbf{A})$ for $f(x) = x^p$.
- The SVD entropy of $\mathbf{A}$ is given by $\mathcal{S}_f(\mathbf{A})$ for $f(x) = x \log x$.
- The log determinant $\log(\det(\mathbf{A}))$ is given by $\mathcal{S}_f(\mathbf{A})$ for $f(x) = \log x$.
- The trace exponential $\mathrm{tr}(\exp(\mathbf{A}))$ is given by $\mathcal{S}_f(\mathbf{A})$ for $f(x) = e^x$.
- The trace inverse $\mathrm{tr}(\mathbf{A}^{-1})$ is given by $\mathcal{S}_f(\mathbf{A})$ for $f(x) = 1/x$.

Applications of Spectral Sums

The applications of approximate computation of the above spectral sums are broad. The log-determinant is important in machine learning and inference applications [Ras04, DKJ+07, FHT08]. The trace inverse is used in uncertainty quantification [BCF09] and quantum chromodynamics [SLO13]. Computation of nuclear norm, is required in a wide variety of applications. It is often used in place of the matrix rank in matrix completion algorithms and other convex relaxations of rank-constrained optimization problems [CR12, DTV11, JNS13, NNS+14]. It appears as the 'graph energy' in theoretical chemistry [Gut92, Gut01], the 'singular value bound' in differential privacy [HLM12, LM12], and in rank aggregation and collaborative ranking [LN15].

Similar to the nuclear norm, general Schatten p-norms are used in convex relaxations for rank-constrained optimization [NHD12, NWC$^+$12]. They also appear in image processing applications such as denoising and background subtraction [XGL$^+$16], classification [LYCG14], restoration [XQT$^+$16], and feature extraction [DHJZ15]. The SVD entropy is used in feature selection [VGLH06, BP14], financial data analysis [Car14, GXL15], and genomic data [ABB00] applications.

Fast Algorithms for Spectral Sum Approximation

Naively, any of the above spectral sums can be computed deterministically and exactly by performing a full SVD of $\mathbf{A}$, which requires $\Omega(n^\omega)$ time in theory and $\Omega(n^3)$ time in practice.[2] The full SVD gives all of $\mathbf{A}$'s singular values $\sigma_1(\mathbf{A}), ..., \sigma_d(\mathbf{A})$ from which the spectral sum can be computed explicitly. Until recently, very few algorithms, randomized or deterministic, with runtimes faster than this approach were known, outside a few special cases of the Schatten p-norms. For example, the squared Frobenius norm ($p = 2$) is trivially computed in $O(\text{nnz}(\mathbf{A}))$ time since it equals the sum of squared entries in $\mathbf{A}$.[3] If randomization along with some small probability of failure are allowed, the Schatten p-norms for *even integers* $p > 2$, or general integers with PSD $\mathbf{A}$ can be approximated in $O(\text{nnz}(\mathbf{A}) \cdot p\epsilon^{-2})$ time via trace estimation [Woo14, BCKY16], since when p is even or $\mathbf{A}$ is PSD, $\mathbf{A}^p$ is PSD and we have $\text{tr}(\mathbf{A}^p) = \|\mathbf{A}\|_p^p$.

In [MNS$^+$18] we use a combination of random sketching methods, stochastic iterative methods, and polynomial approximation to give the first algorithms that break the $\Omega(n^\omega)$ runtime barrier for more general spectral sums, even when randomization and some small probability of failure are allowed. Our main algorithmic result, applying to general Schatten p-norm approximation is:

Theorem 3.1.2 (Theorem 31 of [MNS$^+$18])**.** *For any $p \geq 0$, $\epsilon, \delta > 0$, there is an algorithm that given $\mathbf{A} \in \mathbb{R}^{n \times n}$ returns $X \in [(1-\epsilon)\|\mathbf{A}\|_p^p, (1+\epsilon)\|\mathbf{A}\|_p^p]$ with probability $\geq 1 - \delta$. For $p \geq 2$ the algorithm runs in $\tilde{O}(\log(1/\delta) \cdot p \cdot n^2/\epsilon^3)$ time. For $p < 2$ the*

[2] Note that an exact SVD is uncomputable even with exact arithmetic [TB97]. Nevertheless, direct methods for the SVD obtain superlinear convergence rates and hence are often considered to be 'exact'.

[3] Recall that $\text{nnz}(\mathbf{A})$ denotes the number of nonzero entries in $\mathbf{A}$. See Section 1.3 for definitions of other linear algebraic notation.

algorithm runs in

$$\tilde{O}\left(\log(1/\delta) \cdot \frac{1}{p^3 \cdot \epsilon^{\max\{3,1+1/p\}}} \cdot n^{\frac{2.3729-.0994p}{1+.0435p}}\right)$$

time.

Note that, for $p < 2$, the algorithm of Theorem 3.1.2 uses as a black box the current fastest known matrix multiplication algorithm, which can multiply matrices in $O(n^{\bar{\omega}})$ time for $\bar{\omega} \approx 2.373$ [LG14]. Any improvements in generic matrix multiplication immediately give improvements to the above runtime.

There is also an algorithm that does not use fast matrix multiplication (i.e., uses a $\Theta(n^3)$ time matrix multiplication routine) and runs in

$$\tilde{O}\left(\log(1/\delta) \cdot \frac{1}{p^3 \cdot \epsilon^{\max\{3,1+1/p\}}} \cdot n^{\frac{3+p/2}{1+p/2}}\right)$$

time. For the important case of $p = 1$ (the nuclear norm), Theorem 3.1.2 gives a runtime of $\tilde{O}(n^{2.18}/\epsilon^3)$, or $\tilde{O}(n^{2.33}/\epsilon^3)$ if fast matrix multiplication is not used. We note that our techniques also give algorithms for approximating the SVD entropy and a general class of matrix Orlicz norms. We refer the reader to [MNS$^+$18] for details.

Note that if ϵ is a constant, both the runtimes shown above are faster than the fastest known matrix multiplication algorithm ($O(n^{\bar{\omega}})$ for $\bar{\omega} \approx 2.373$ [LG14]). Thus, they are faster than the naive approach of performing a full SVD. However, for small ϵ, the runtime dependence on $1/\epsilon^3$ can be very expensive. It is natural to ask:

Question 3.1.3. *Is possible to achieve the best of both worlds: a highly accurate spectral sum algorithm (e.g., with* $\log(1/\epsilon)$ *dependence), but with an* $o(n^{\bar{\omega}})$ *runtime, where* $\bar{\omega}$ *is the lowest known exponent for fast matrix multiplication?* [4]

3.1.2 Kernel Low-Rank Approximation

The kernel method is a popular technique used to apply linear learning and classification algorithms to datasets with nonlinear structure [SS02, STC04]. Given training input points $\mathbf{a}_1, ..., \mathbf{a}_n \in \mathbb{R}^d$, the idea is to replace the standard Euclidean dot product $\langle \mathbf{a}_i, \mathbf{a}_j \rangle = \mathbf{a}_i^T \mathbf{a}_j$ with the kernel dot product $\psi(\mathbf{a}_i, \mathbf{a}_j)$, where $\psi : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^+$ is some positive semidefinite function. Popular kernel functions include e.g., the Gaussian kernel with $\psi(\mathbf{a}_i, \mathbf{a}_j) = e^{-\|\mathbf{a}_i - \mathbf{a}_j\|^2/\sigma}$ for some parameter σ and the polynomial kernel of degree q with $\psi(\mathbf{a}_i, \mathbf{a}_j) = (c + \mathbf{a}_i^T \mathbf{a}_j)^q$ for some parameter c.

[4] See Section 1.3 for a formal definition of $\bar{\omega}$ and the related quantity ω.

In this chapter, we focus on kernels where $\psi(\mathbf{a}_i, \mathbf{a}_j)$ is a function of the dot products $\mathbf{a}_i^T\mathbf{a}_i = \|\mathbf{a}_i\|^2$, $\mathbf{a}_j^T\mathbf{a}_j = \|\mathbf{a}_j\|^2$, and $\mathbf{a}_i^T\mathbf{a}_j$. Such functions encompass many kernels used in practice, including the Gaussian kernel, the Laplace kernel, the polynomial kernel, and the Matern kernels.

Any positive semidefinite function $\psi(\cdot, \cdot)$ can be associated with a reproducing kernel Hilbert space $\mathcal{F}$ such that $\psi(\mathbf{a}_i, \mathbf{a}_j) = \langle \phi(\mathbf{a}_i), \phi(\mathbf{a}_j)\rangle_{\mathcal{F}}$ where $\phi : \mathbb{R}^d \rightarrow \mathcal{F}$ is a typically non-linear *feature map*. We let $\mathbf{\Phi} = [\phi(\mathbf{a}_1), ..., \phi(\mathbf{a}_n)]^T$ denote the kernelized dataset, whose i^{th} row is the kernelized datapoint $\phi(\mathbf{a}_i)$.

There is no requirement that $\mathbf{\Phi}$ can be efficiently computed or stored – for example, in the case of the Gaussian kernel, $\mathcal{F}$ is an infinite dimensional space. Thus, kernel methods typically work with the kernel matrix $\mathbf{K} \in \mathbb{R}^{n\times n}$ with $\mathbf{K}_{i,j} = \psi(\mathbf{a}_i, \mathbf{a}_j)$. We will also sometimes denote $\mathbf{K} = \{\psi(\mathbf{a}_i, \mathbf{a}_j)\}$ to make it clear which kernel function it is generated by. We can equivalently write $\mathbf{K} = \mathbf{\Phi}\mathbf{\Phi}^T$. As long as all operations of an algorithm only access $\mathbf{\Phi}$ via the dot products between its rows, they can thus be implemented using just $\mathbf{K}$ without explicitly computing the feature map.

Unfortunately computing $\mathbf{K}$ is expensive, and a bottleneck for scaling kernel methods to large datasets. For the kernels we consider, where ψ depends on dot products between the input points, we must at least compute the Gram matrix $\mathbf{A}\mathbf{A}^T$, requiring $\Theta(n^2 d)$ time in general. Even if $\mathbf{A}$ is sparse, this takes $\Theta(\text{nnz}(\mathbf{A})n)$ time. Storing $\mathbf{K}$ then takes $\Theta(n^2)$ space, and processing it for downstream applications like kernel ridge regression and kernel SVM can be even more expensive.

Low-Rank Kernel Approximation

For this reason, a vast body of work studies how to efficiently approximate $\mathbf{K}$ via a low-rank surrogate $\tilde{\mathbf{K}}$ [SS00, AMS01, WS01, FS02, RR07, ANW14, LSS13, BJ02, DM05, ZTK08, BW09, CKS11, WZ13, GM13]. If $\tilde{\mathbf{K}}$ is rank-k, it can be stored in factored form in $O(nk)$ space and operated on quickly – e.g., it can be inverted in just $O(nk^{\bar{\omega}-1})$ time to solve kernel ridge regression.

One possibility is to set $\tilde{\mathbf{K}} = \mathbf{K}_k$ where $\mathbf{K}_k$ is $\mathbf{K}$'s best k-rank approximation – the projection onto its top k eigenvectors. $\mathbf{K}_k$ minimizes, over all rank-k $\tilde{\mathbf{K}}$, the error $\|\mathbf{K} - \tilde{\mathbf{K}}\|_F$, recalling that $\|\mathbf{M}\|_F$ is the Frobenius norm: $\left(\sum_{i,j}\mathbf{M}_{i,j}^2\right)^{1/2}$. It in fact minimizes error under any unitarily invariant norm, e.g., the popular spectral norm. Unfortunately, $\mathbf{K}_k$ is prohibitively expensive to compute, requiring $\Omega(n^\omega)$ time even using fast matrix multiplication

Research on low-rank kernel approximation thus seeks to find $\tilde{\mathbf{K}}$ that is nearly as good as $\mathbf{K}_k$, but can be computed much more quickly. Specifically, it is natural to ask for $\tilde{\mathbf{K}}$ fulfilling the following near optimal low-rank approximation guarantee of Problem 2.1.1 in Chapter 2, restated here:

Definition 3.1.4 (Relative Error Kernel Approximation)**.** *For any $\epsilon \geq 0$, $k \in \mathbb{Z}^{\geq 1}$, $\tilde{\mathbf{K}}$ is a $(1+\epsilon)$ relative error rank-k approximation of $\mathbf{K}$ if:*

$$\|\mathbf{K} - \tilde{\mathbf{K}}\|_F \leq (1+\epsilon)\|\mathbf{K} - \mathbf{K}_k\|_F. \tag{3.1}$$

Other goals, such as nearly matching the spectral norm error $\|\mathbf{K} - \mathbf{K}_k\|_2$ or approximating $\mathbf{K}$ entry-wise have also been considered [RR07, GM13]. Of particular interest to our results is the closely related goal:

Definition 3.1.5 (Approximate Kernel PCA)**.** *For any $\epsilon > 0$, an orthonormal basis $\mathbf{Z} \in \mathbb{R}^{n \times k}$ is a $(1+\epsilon)$ approximation solution to rank-k kernel PCA if, for any $\mathbf{\Phi}$ with $\mathbf{\Phi}\mathbf{\Phi}^T = \mathbf{K}$:*

$$\|\mathbf{\Phi} - \mathbf{Z}\mathbf{Z}^T\mathbf{\Phi}\|_F \leq (1+\epsilon)\|\mathbf{\Phi} - \mathbf{\Phi}_k\|_F. \tag{3.2}$$

Definition 3.1.5 asks us to find a low-rank subspace $\mathbf{Z}$ such that the projection of our kernelized dataset $\mathbf{\Phi}$ onto $\mathbf{Z}$ nearly optimally approximates this dataset. Given $\mathbf{Z}$, we can approximate $\mathbf{K}$ using $\tilde{\mathbf{K}} = \mathbf{Z}\mathbf{Z}^T\mathbf{\Phi}\mathbf{\Phi}^T\mathbf{Z}\mathbf{Z}^T = \mathbf{Z}\mathbf{Z}^T\mathbf{K}\mathbf{Z}\mathbf{Z}^T$. Alternatively, letting $\mathbf{P}$ be the projection onto the row span of $\mathbf{Z}\mathbf{Z}^T\mathbf{\Phi}$, we can write $\tilde{\mathbf{K}} = \mathbf{\Phi}\mathbf{P}\mathbf{\Phi}^T$, which can be computed efficiently, for example, when $\mathbf{P}$ is a projection onto a subset of the kernelized datapoints [MM17].

Fast Algorithms for Relative-Error Kernel Approximation

Until recently, all algorithms achieving the guarantees of Definitions 3.1.4 and 3.1.5 were at least as expensive as computing the full matrix $\mathbf{K}$, which was needed to compute the low-rank approximation [GM13].

However, recent work has shown that this is not required. Avron, Nguyen, and Woodruff [ANW14] demonstrate that for the polynomial kernel, $\mathbf{Z}$ satisfying Definition 3.1.5 can be computed in $O(\text{nnz}(\mathbf{A})q) + n\,\text{poly}(3^q k/\epsilon)$ time for a polynomial kernel with degree q.

In [MM17] we give a fast algorithm for *any kernel*, using recursive Nyström sampling, which outputs $\mathbf{Z}$ satisfying Definition 3.1.5 (see Section C.3 of [MM17]). Computing $\mathbf{Z}$ requires accessing $\tilde{O}(k/\epsilon)$ columns of the kernel matrix along with $\tilde{O}(n(k/\epsilon)^{\bar{\omega}-1})$ additional time for other computations. Assuming the kernel is a function of the dot products between the input points, the kernel evaluations require

$\tilde{O}(\text{nnz}(\mathbf{A})k/\epsilon)$ time. The results of [MM17] can also be used to compute $\tilde{\mathbf{K}}$ satisfying Definition 3.1.4 with $\epsilon = \sqrt{n}$ in $\tilde{O}(\text{nnz}(\mathbf{A})k + nk^{\bar{\omega}-1})$ time (see detailed discussion in Chapter 2.)

The low-rank approximation results of Chapter 2 apply to any PSD matrix, and therefore, to any kernel matrix. Thus, as a consequence of Theorem 2.5.1, for any kernel and any $\epsilon > 0$, it is possible to compute $\tilde{\mathbf{K}}$ satisfying Definition 3.1.4 with probability $\geq 99/100$ in $\tilde{O}(\text{nnz}(\mathbf{A})k/\epsilon) + n\,\text{poly}(k/\epsilon)$ time plus the time needed to compute an $\tilde{O}(\sqrt{nk}/\epsilon^2) \times \tilde{O}(\sqrt{nk/\epsilon})$ submatrix of $\mathbf{K}$. If $\mathbf{A}$ has uniform row sparsity – i.e., $\text{nnz}(\mathbf{a}_i) \leq c\,\text{nnz}(\mathbf{A})/n$ for some constant c and all i, this step can be done in $\tilde{O}(\text{nnz}(\mathbf{A})k/\epsilon^{2.5})$ time. Alternatively, if $d \leq (\sqrt{nk}/\epsilon^2)^\alpha$ for $\alpha < .314$ this can be done in $\tilde{O}(nk/\epsilon^4) = \tilde{O}(\text{nnz}(\mathbf{A})k/\epsilon^4)$ time using fast rectangular matrix multiplication [LG12, GU17] (assuming that there are no all zero data points so $n \leq \text{nnz}(\mathbf{A})$.)

As discussed, the algorithms of [MM17, MW17b] make significant progress in efficiently solving the low-rank approximation problems of Definitions 3.1.4 and 3.1.5 for general kernel matrices. They demonstrate that, surprisingly, a relative-error low-rank approximation can be computed significantly faster than the time required to write down all of $\mathbf{K}$, which is $\Omega(\text{nnz}(\mathbf{A}) \cdot n)$ for kernels that depend on the dot products between the input points. It is natural to ask:

Question 3.1.6. *Can the results of [MM17, MW17b] be improved significantly? Even ignoring ϵ dependencies and typically lower order terms, both algorithms use $\Omega(\text{nnz}(\mathbf{A})k)$ time. One might hope to improve this to input sparsity, or near input sparsity time, $\tilde{O}(\text{nnz}(\mathbf{A}))$, which is known for computing a low-rank approximation of $\mathbf{A}$ itself [CW13]. The work of Avron et al. affirms that this is possible for the kernel PCA guarantee of Definition 3.1.5 for degree-q polynomial kernels, for constant q. Can this result be extended to other popular kernels, or even more general classes?*

3.1.3 Our Contributions

In this chapter we show that general matrix multiplication can be reduced to very accurate spectral sum approximation as well as to relative error kernel low-rank approximation (in the sense of Definition 3.1.4), for any error parameter ϵ. These reductions help pin down the complexity of these basic problems. Our main results are described below.

Spectral Sum Problems

We show that any algorithm (randomized or deterministic) for computing a sufficiently accurate approximation to any of the important spectral sums defined in Section 3.1.1 can be used to exactly solve the *triangle detection problem*: decide if an n-node graph contains at least one triangle [IR77, CN85]. Our reductions all follow from a single general theorem (Theorem 3.2.1) and are deterministic. As an example of one of our results, for the Schatten 3-norm we prove in Corollary 3.2.3:

Theorem 3.1.7 (Schatten 3-Norm Hardness)**.** *Suppose there exists an algorithm that on any input $\mathbf{B} \in \mathbb{R}^{n\times n}$ returns, with probability $1-\delta$, $X \in [(1-\epsilon)\|\mathbf{B}\|_3^3, (1+\epsilon)\|\mathbf{B}\|_3^3]$ in $O(n^\gamma \epsilon^{-c})$ time. Then one can solve triangle detection on n-node graphs with success probability $1-\delta$ in $O(n^{\gamma+4c})$ time.*

The triangle detection problem is a canonical problem in fine-grained complexity. Generally, it is believed that triangle detection on a general n-node graph is as difficult as multiplying two $n \times n$ matrices together. Naively this requires $\Omega(n^3)$ time. Using fast matrix multiplication it requires $\Omega(n^\omega)$ time. No faster runtimes are known even if randomization and a small probability of failure are allowed.

In the seminal work of [WW10] it was shown that any truly subcubic time algorithm (i.e., an algorithm running in $O(n^{3-c})$ time for some $c > 0$) for triangle detection yields a truly subcubic time algorithm for Boolean matrix multiplication (BMM), for which, like general matrix multiplication, the fastest runtime is $\Omega(n^\omega)$. This result is via a deterministic reduction from BMM to triangle reduction. Formally the result can be stated as:

Theorem 3.1.8 (Implied by Lemma E.1 of [WW10])**.** *For any $0 < c \leq 1$, suppose there is a $O(n^{3-c})$ time algorithm for triangle detection on n-node graphs with success probability $1-\delta$. Then there is an $O(n^{3-c/3})$ time algorithm for Boolean matrix multiplication with success probability $1 - O(\delta \cdot n^2 \log n)$.*[5]

Consequently, our results show that approximating any of the spectral sums above to high enough accuracy is in a sense as difficult as exact BMM. For example, Theorem 3.1.2 gives an algorithm for the Schatten 3-norm that succeeds with high probability and requires just $\tilde{O}(n^2/\epsilon^3)$ time for general matrices. In comparison, currently

[5] Note that in the reduction Theorem 3.1.8 the triangle detection algorithm is called $O(n^2 \log n)$ times, and so the probability of failure may increase correspondingly. However, for spectral sum computation and triangle detection, the runtime dependence on the failure probability δ is always at worst $O(\log 1/\delta)$, so this loss translates to a $O(\log n)$ loss in runtime. This is because it is always possible to solve the problem with success probability $\delta' = 2/3$. Repeating this $O(\log 1/\delta)$ times and taking the median of the outputs will return a valid solution with probability $1-\delta$.

the fastest known matrix multiplication algorithm runs in $O(n^{\bar{\omega}})$ time for $\bar{\omega} \approx 2.373$ [LG14]. By Theorem 3.1.7, improving the ϵ dependence to $o(1/\epsilon^{(\bar{\omega}-2)/4}) = O(1/\epsilon^{.09})$, would yield a randomized algorithm for triangle detection for general graphs succeeding with high probability and running faster than matrix multiplication time, breaking a longstanding runtime barrier for this problem. Even a $1/\epsilon^{1/3}$ dependence would give a sub-cubic time triangle detection algorithm, and hence a subcubic time matrix multiplication algorithm via the reduction of Theorem 3.1.8. This would represent a major breakthrough in linear algebra as it would give an alternative approach to fast matrix multiplication. Thus, we show that the answer to Question 3.1.3 is likely no, barring a major algorithmic breakthrough.

We extend our lower bound for $\text{tr}(\mathbf{A}^{-1})$ by proving that approximating $\text{tr}(\mathbf{A}^{-1})$ for the $\mathbf{A}$ used in the lower bound can be reduced to computing all effective resistances of a certain graph Laplacian up to $(1 \pm \epsilon)$ error. Thus, we rule out highly accurate (with $1/\epsilon^c$ dependence for small c) fast approximation algorithms for all effective resistances, barring a major breakthrough in the state-of-the-art in triangle detection. Interestingly, this bound holds despite the existence of nearly linear time system solvers (with $\log(1/\epsilon)$ error dependence) for Laplacians [ST04].

The effective resistances of a graph Laplacian are just scalings of the leverage scores of the corresponding vertex edge incidence matrix. As discussed in Chapters 1 and 2, effective resistances and leverage scores have been crucial in giving algorithmic improvements to fundamental problems like graph sparsification [SS08], regression [LMP13, CLM+15], and low-rank approximation [CMM17, MW17b]. While coarse multiplicative approximations to the quantities suffice for these problems, more recently computing these quantities has been used to achieve breakthroughs in solving maximum flow and linear programming [LS14], cutting plane methods [LSW15], and sampling random spanning trees [KM09, MST15]. In each of these settings having more accurate estimates would be a natural route to either simplify or possibly improve existing results; our results show that this is unlikely to be successful if the precision requirements are two high.

Kernel Low-Rank Approximation

We show that achieving the guarantee of Definition 3.1.4 significantly more efficiently than the work of [MM17, MW17b] is likely very difficult (i.e., the answer to Question 3.1.6 is likely no). Specifically, we prove that for a wide class of kernels, the kernel low-rank approximation problem is as hard as multiplying the input $\mathbf{A} \in \mathbb{R}^{n \times d}$ by an arbitrary $\mathbf{C} \in \mathbb{R}^{d \times k}$. This cannot be done in $o(\text{nnz}(\mathbf{A})k)$ time in general, and giving an algorithm to do so would represent a major breakthrough in fast matrix multiplication. Similarly to our work on spectral sums, our lower bound is via a reduction from *exact* rectangular matrix multiplication to *approximate* kernel low-rank approximation. We have the following result for some common kernels to which our techniques apply:

Theorem 3.1.9 (Hardness for low-rank kernel approximation)**.** *Consider any polynomial kernel* $\psi(\mathbf{m}_i, \mathbf{m}_j) = (c+\mathbf{m}_i^T\mathbf{m}_j)^q$*, Gaussian kernel* $\psi(\mathbf{m}_i, \mathbf{m}_j) = e^{-\|\mathbf{m}_i-\mathbf{m}_j\|^2/\sigma}$*, or the linear kernel* $\psi(\mathbf{m}_i, \mathbf{m}_j) = \mathbf{m}_i^T\mathbf{m}_j$*. Assume there is an algorithm that, for some* $\delta \geq 0$ *and any* n, d, k*: for some approximation factor* Δ*, given* $\mathbf{M} \in \mathbb{R}^{n\times d}$ *with associated kernel matrix* $\mathbf{K} = \{\psi(\mathbf{m}_i, \mathbf{m}_j)\}$*, in* $o(\text{nnz}(\mathbf{M})k + nk^p)$ *time, for* $p \geq 2$*, returns* $\mathbf{N} \in \mathbb{R}^{n\times k}$ *that satisfies with probability* $1 - \delta$*:*

$$\|\mathbf{K} - \mathbf{N}\mathbf{N}^T\|_F^2 \leq \Delta\|\mathbf{K} - \mathbf{K}_k\|_F^2.$$

Then, for any n, d, k*, there is an* $o(\text{nnz}(\mathbf{A})k+nk^c)$ *time algorithm that, given arbitrary* $\mathbf{A} \in \mathbb{Z}^{n\times d}$*,* $\mathbf{C} \in \mathbb{Z}^{d\times k}$*, returns their product* $\mathbf{AC}$ *with probability* $\geq 1 - \delta$*.*

The above applies for *any approximation factor* Δ. While we work in the real RAM model, ignoring bit complexity, as long as $\Delta = \text{poly}(n)$ and $\mathbf{A}, \mathbf{C}$ have entries bounded by some polynomial in n and d, our reduction from multiplication to low-rank approximation is achieved using matrices that can be represented with just $O(\log(n + d))$ bits per entry.

Theorem 3.1.9 shows that the runtime of $\tilde{O}(\text{nnz}(\mathbf{A})k+nk^{\bar{\omega}-1})$ for $\Delta = \sqrt{n}$ achieved by [MM17] for general kernels cannot be significantly improved without advancing the state-of-the-art in matrix multiplication. The fastest known algorithm for performing this multiplication, even if randomization and some probability of failure are allowed, runs in time $\Omega(\text{nnz}(\mathbf{A})k)$ for sufficiently sparse $\mathbf{A}$ (see [LG12, GU17] for details.)

As discussed, when $\mathbf{A}$ has uniform row sparsity or when $d \leq (\sqrt{nk}/\epsilon^2)^\alpha$, the runtime of [MW17b] for $\Delta = (1 + \epsilon)$, ignoring ϵ dependencies and typically lower order terms, is $\tilde{O}(\text{nnz}(\mathbf{A})k)$, which is also nearly tight (see Chapter 2 for a detail exposition of this result).

In recent work, Backurs et al. [BIS17] give conditional lower bounds for a number of kernel learning problems, including kernel PCA for the Gaussian kernel. However, their strong bound, of $\Omega(n^2)$ time, requires very small error $\Delta = \exp(-\omega(\log^2 n)$, whereas ours applies for any relative error Δ.

In [MW17a] we give an algorithm which shows that, in contrast, for the Kernel PCA guarantee of Definition 3.1.5, it is possible to obtain $o(\text{nnz}(\mathbf{A})k)$ time randomized algorithms for any shift and rotationally invariant kernel – e.g., any radial basis function kernel where $\psi(\mathbf{x}_i, \mathbf{x}_j) = f(\|\mathbf{x}_i - \mathbf{x}_j\|)$. This result significantly extends the progress of Avron et al. [ANW14] on the polynomial kernel. We refer the reader to [MW17a] for details.

3.1.4 Prior Work

Fine-grained complexity has had much success for graph problems, string problems, and problems in other areas (see, e.g., [Wil15] for a survey), and is closely tied to understanding the complexity of matrix multiplication. However, to the best of our knowledge it has not been applied broadly to problems in linear algebra. [BIS17] gives one example of work similar to our own which uses fine-grained complexity approaches to give conditional lower bounds for a number of problems in machine learning, including the linear algebraic problems of kernel matrix approximation and kernel ridge regression.

Outside this example, hardness results for linear algebraic problems tend to apply to restricted computational models such as arithmetic circuits [BS83], bilinear circuits or circuits with bounded coefficients and number of divisions [Mor73, Raz03, Shp03, RS03], algorithms for linear systems that can only add multiples of rows to each other [KKS65, KS70], and algorithms with restrictions on the dimension of certain manifolds defined in terms of the input [Win70, Win87, Dem13]. In contrast, we obtain conditional lower bounds for arbitrary polynomial time algorithms by showing that faster algorithms for them imply faster algorithms for canonical hard problems in fine-grained complexity – specifically, for square and rectangular matrix multiplication.

3.2 Lower Bounds for Spectrum Approximation

In this section we prove our conditional lower bounds for spectral sums, including the Schatten p-norms, log-determinant, the SVD entropy, the trace inverse, and the trace exponential. We show that similar techniques can also be used to give hardness for computing the determinant. We also show that our bounds imply hardness for the important primitives of computing effective resistances in a graph or leverage scores in a matrix.

Road Map: In Section 3.2.1 we give a high level overview of our lower bound approach. In Section 3.2.2 we give our general result on reducing triangle detection to spectral sum computation. In Section 3.2.3 we use this result to show hardness for computing various well studied spectral sums. In Section 3.2.4 we extend our results to give hardness for graph effective resistances and leverage scores.

3.2.1 Lower Bound Approach

As discussed in Section 3.1.3, our conditional lower bounds reduce triangle detection to spectral sum approximation. Boolean matrix multiplication can further be reduced to triangle detection by Theorem 3.1.8., thereby giving a reduction from exact Boolean matrix multiplication to spectral sum approximation. Our main reduction will be for a very general class of spectral sums. We will then show that this class encompasses the specific sums that we care about.

To illustrate the main idea of our reduction, it is instructive to first consider just the Schatten 3-norm, $\|\mathbf{A}\|_3^3 = S_f(\mathbf{A})$ for $f(x) = x^3$. Our general reduction will be based off this case. We start with the fact that the number of triangles in any unweighted graph G with n nodes is equal to $\operatorname{tr}(\mathbf{A}^3)/6$, where $\mathbf{A} \in \mathbb{R}^{n\times n}$ is the adjacency matrix of G. Any algorithm for approximating $\operatorname{tr}(\mathbf{A}^3)$ to high enough accuracy therefore gives an algorithm for detecting if a graph has at least one triangle.

For any $\mathbf{A}$ we can write

$$\operatorname{tr}(\mathbf{A}^3) = \sum_{i=1}^{n} \lambda_i(\mathbf{A})^3,$$

where $\lambda_i(\mathbf{A})$ is $\mathbf{A}$'s i^{th} eigenvalue. Unfortunately, since $\mathbf{A}$ is not PSD, its eigenvalues may have different signs than its singular values and so $\operatorname{tr}(\mathbf{A}^3)$ is not simply a function of $\mathbf{A}$'s singular values and cannot be determined simply via a spectral sum computation. However, the graph Laplacian given by $\mathbf{L} = \mathbf{D} - \mathbf{A}$ where $\mathbf{D}$ is the diagonal degree matrix, is PSD. This means that its eigenvalues $\lambda_1(\mathbf{L}), ..., \lambda_n(\mathbf{L})$ are equal to its singular values $\sigma_1(\mathbf{L}), ..., \sigma_n(\mathbf{L})$. Again using that $\operatorname{tr}(\mathbf{M}) = \sum_{i=1}^n \lambda_i(\mathbf{M})$ for any $\mathbf{M}$ we have:

$$\begin{aligned}
\|\mathbf{L}\|_3^3 &= \sum_{i=1}^{n} \sigma_i(\mathbf{L})^3 \\
&= \sum_{i=1}^{n} \lambda_i(\mathbf{L})^3 \\
&= \operatorname{tr}(\mathbf{L}^3) \\
&= \operatorname{tr}(\mathbf{D}^3) - 3\operatorname{tr}(\mathbf{D}^2\mathbf{A}) + 3\operatorname{tr}(\mathbf{D}\mathbf{A}^2) - \operatorname{tr}(\mathbf{A}^3).
\end{aligned}$$

We have $\text{tr}(\mathbf{D}^2\mathbf{A}) = 0$ since $\mathbf{A}$ has an all 0 diagonal. Further, it is not hard to see that $\text{tr}(\mathbf{D}\mathbf{A}^2) = \text{tr}(\mathbf{D}^2)$. So this term and $\text{tr}(\mathbf{D}^3)$ are easy to compute exactly in $O(n^2)$ time – they are simple functions of the node degrees in the graph. Thus, if we approximate $\|\mathbf{L}\|_3^3$ up to additive error 6, we can determine if $\text{tr}(\mathbf{A}^3) = 0$ or $\text{tr}(\mathbf{A}^3) \geq 6$ and so detect if G contains a triangle. One can show that $\|\mathbf{L}\|_3^3 \leq 8n^4$ for any unweighted graph with n nodes, and hence computing this norm up to $(1 \pm \epsilon)$ relative error for $\epsilon = 3/(6n^4)$ suffices to detect a triangle. If we have an $O(n^\gamma \epsilon^{-c})$ time $(1 \pm \epsilon)$ approximation algorithm for the Schatten 3-norm, we can thus perform triangle detection in $O(n^{\gamma+4c} + n^2)$ time. The n^2 term is required for the exact computation of $\text{tr}(\mathbf{D}\mathbf{A}^2)$ and $\text{tr}(\mathbf{D}^3)$. We typically imagine that it is dominated by the $O(n^{\gamma+4c})$ term since, without any assumptions on $\mathbf{A}$, computing the Schatten 3-norm at least requires reading all of $\mathbf{A}$ which takes $\Omega(n^2)$ time.

Generalizing to Other Spectral Sums

We can generalize the above approach to the Schatten 4-norm by adding λ self-loops to each node of G, which corresponds to replacing $\mathbf{A}$ with $\lambda\mathbf{I} + \mathbf{A}$. We then consider $\text{tr}((\lambda\mathbf{I} + \mathbf{A})^4) = \|\lambda\mathbf{I} + \mathbf{A}\|_4^4$. This is the sum over all vertices v_i for $i \in [n]$ of the number of paths that start at v_i and return to v_i in four steps. All of these paths are either (1) legitimate four cycles that exist in G, (2) triangles combined with self loops, or (3) combinations of self-loops and two-step paths from a vertex v_i to one of its neighbors and back. The number of type (3) paths is exactly computable using the node degrees and number of self loops. Additionally, if the number of self loops λ is large enough, the number of type (2) paths will dominate the number of type (1) paths, even if there is just a single triangle in the graph. Hence, an accurate approximation to $\|\lambda\mathbf{I} + \mathbf{A}\|_4^4$ will give us the number of type (2) paths, from which we can easily compute the number of triangles.

A similar argument extends to a very broad class of spectral sums by considering a power series expansion of $f(x)$ and showing that for large enough λ, $\text{tr}\left(f(\lambda\mathbf{I} + \mathbf{A})\right)$ is dominated by $\text{tr}(\mathbf{A}^3)$ along with some exactly computable terms. Thus, an accurate approximation to this spectral sum allows us to determine the number of triangles in G. This approach works for any $f(x)$ that can be represented as a power series, with reasonably well-behaved coefficients on some interval of $\mathbb{R}^+$, giving our bounds for all $\|\mathbf{A}\|_p$ with $p \neq 2$, the SVD entropy, $\log\det(\mathbf{A})$, $\text{tr}(\mathbf{A}^{-1})$, and $\text{tr}(\exp(\mathbf{A}))$. In Figure **??** we illustrate the approach as applied to $\text{tr}(\mathbf{A}^{-1})$.

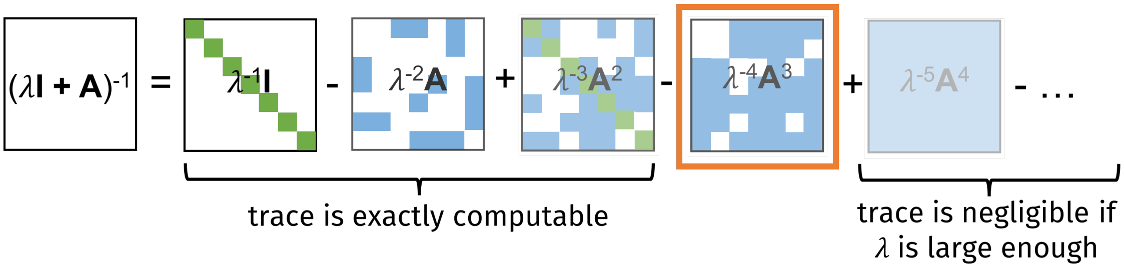
(λI + A)-1
=
λ-1I
-
λ-2A
+
λ-3A2
-
λ-4A3
+
λ-5A4
- …
trace is exactly computable
trace is negligible if λ is large enough

Figure 3-1: An illustration of our lower bound technique applied to the trace inverse. By writing out the Taylor expansion $\frac{1}{\lambda+x} = \sum_{i=0}^{\infty} \frac{x^i}{\lambda^{i+1}}$ we can see that for sufficiently large λ, $\text{tr}((\lambda\mathbf{I}+\mathbf{A})^{-1})$ is dominated by $\text{tr}(\lambda^{-4}\mathbf{A}^3)$ along with some exactly computable terms. Thus, computing $\text{tr}((\lambda\mathbf{I}+\mathbf{A})^{-1})$ to high enough accuracy lets us approximate $\text{tr}(\mathbf{A}^3)$ and in turn detect a triangle in the graph.

3.2.2 Reductions From Triangle Detection

Here we provide our main technical tool for reducing triangle detection to spectral sum computation. As discussed in Section 3.2.1, our reduction leverages the well known fact that the number of triangles in any unweighted graph G is equal to $\text{tr}(\mathbf{A}^3)/6$ where $\mathbf{A}$ is the adjacency matrix for G. Consequently, given any function $f : \mathbb{R}^+ \to \mathbb{R}^+$ whose power series is reasonably behaved, we can show that for suitably small Δ the quantity $\text{tr}(f(\mathbf{I}+\Delta\mathbf{A}))$ is dominated by the contribution of $\text{tr}(\mathbf{A}^k)$ for $k \in (0,3)$. Therefore computing $\text{tr}(f(\mathbf{I}+\Delta\mathbf{A}))$ approximately lets us distinguish between whether or not $\text{tr}(\mathbf{A}^3) = 0$ or $\text{tr}(\mathbf{A}^3) \geq 6$, and therefore detect a triangle in G.

We formalize this in the following theorem. As it simplifies the result, we focus on the case where f can be written as a power series on the interval $(0,2)$. This suffices for our purposes and can be generalized via shifting and scaling of x.

Theorem 3.2.1 (Many Spectral Sums are as Hard as Triangle Detection)**.** *Let $f : \mathbb{R}^+ \to \mathbb{R}^+$ be an arbitrary function, such that for $x \in (0,2)$ we can express it as*

$$f(x) = \sum_{k=0}^{\infty} c_k (x-1)^k \textit{ where } \left|\frac{c_k}{c_3}\right| \leq h^{k-3} \textit{ for all } k > 3. \tag{3.3}$$

Then given the adjacency matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$ of any simple graph G that has no self-loops and spectral sum estimate

$$X \in [(1-\epsilon_1)\sum_{i=1}^{n} f(\sigma_i(\mathbf{I}-\Delta\mathbf{A})), (1+\epsilon_1)\sum_{i=1}^{n} f(\sigma_i(\mathbf{I}-\Delta\mathbf{A}))]$$

for scaling Δ and accuracy ϵ_1 satisfying

$$\Delta = \min\left\{\frac{1}{n}\ ,\ \frac{1}{10n^4h}\right\} \text{ and } \epsilon_1 = \frac{1}{9}\cdot\min\left\{1\ ,\ \left|\frac{c_3\Delta^3}{c_0 n}\right|\ ,\ \left|\frac{c_3\Delta}{c_2 n^2}\right|\right\}$$

we can detect if G has a triangle in $O(\mathrm{nnz}(\mathbf{A})) = O(n^2)$ time.

Theorem 3.2.1 shows that, if we are given a $(1\pm\epsilon_1)$ approximation to $f(\mathbf{I}-\Delta\mathbf{A})$ for appropriately set ϵ_1, Δ, then we can detect if G contains a triangle. This immediately implies that if we have an algorithm for computing such an approximation with probability $\geq 1-\delta$, then we can use it to perform triangle detection with success probability $\geq 1-\delta$. Formally:

Corollary 3.2.2. *Let f and ϵ_1 be as in Theorem 3.2.1. Assume there exists an algorithm that, for some $\delta \geq 0$, on any input $\mathbf{A} \in \mathbb{R}^{n\times n}$, $\epsilon > 0$, with probability $\geq 1-\delta$ returns:*

$$X \in [(1-\epsilon)\mathcal{S}_f(\mathbf{A}), (1+\epsilon)\mathcal{S}_f(\mathbf{A})]$$

in $O(g(n, \mathrm{nnz}(\mathbf{A}))\cdot\epsilon^{-c})$ time. Then one can solve triangle detection on n-node graphs with success probability $\geq 1-\delta$ in $O(\mathrm{nnz}(\mathbf{A}) + g(n,\mathrm{nnz}(\mathbf{A}))\cdot\epsilon_1^{-c}))$ time.

Proof. For any graph G with adjacency matrix $\mathbf{A}$, we can run the assumed algorithm in $O(g(n,\mathrm{nnz}(\mathbf{A}))\cdot\epsilon_1^{-c})$ time to compute an approximation X of $\mathcal{S}_f(\mathbf{A})$. If $X \in [(1-\epsilon)\mathcal{S}_f(\mathbf{A}), (1+\epsilon)\mathcal{S}_f(\mathbf{A})]$, by Theorem 3.2.1 we can detect if any G has a triangle in $O(\mathrm{nnz}(\mathbf{A}))$ time. By assumption, $X \in [(1-\epsilon)\mathcal{S}_f(\mathbf{A}), (1+\epsilon)\mathcal{S}_f(\mathbf{A})]$ with probability $\geq 1-\delta$. Thus our algorithm for triangle detection succeeds with probability $\geq 1-\delta$ and runs in time $O(\mathrm{nnz}(\mathbf{A}) + g(n,\mathrm{nnz}(\mathbf{A}))\cdot\epsilon_1^{-c}))$. □

We now prove Theorem 3.2.1. As discussed in Section 3.2.1, we will show that it is possible to extract an accurate approximation of $\mathrm{tr}(\mathbf{A}^3)$, which is proportional to the number of triangles in G, from the assumed approximation to $\mathcal{S}_f(\mathbf{I}-\Delta\mathbf{A})$. This will be possible by our assumption that in the series expansion $f(x) = \sum_{k=0}^{\infty} c_k(x-1)^k$, the coefficients c_k for $k > 3$ are not too large compared to c_3.

Proof of Theorem 3.2.1 . Let $\mathbf{A}$, G, Δ, ϵ_1, and X be as in the theorem statement and let $\mathbf{B} \overset{\text{def}}{=} \mathbf{I}-\Delta\mathbf{A}$. By Gershgorin's circle theorem [Wei03], $\|\mathbf{A}\|_2 \leq n-1$ and since $\Delta \leq 1/n$, $\|\Delta\mathbf{A}\|_2 < 1$. Consequently $\mathbf{B}$ is symmetric PSD with $\sigma_i(\mathbf{B}) = \lambda_i(\mathbf{B}) \in (0,2)$

for all $i \in [n]$. Therefore, using (3.3) yields:

$$\begin{aligned}\sum_{i=1}^{n} f(\sigma_i(\mathbf{B})) &= \sum_{i=1}^{n} f(1-\Delta\lambda_i(\mathbf{A}))\\ &= \sum_{i=1}^{n}\sum_{k=0}^{\infty} c_k(\Delta\lambda_i(\mathbf{A}))^k\\ &= \sum_{k=0}^{\infty} c_k\Delta^k \operatorname{tr}(\mathbf{A}^k)\,. \qquad (3.4)\end{aligned}$$

Setting $\Delta \leq \frac{1}{10n^4h}$ is enough to insure that the first four terms of this power series dominate. Since we can compute the first three terms exactly, this will allow us to obtain a good approximation to the fourth term, which is a multiple of $\operatorname{tr}(\mathbf{A}^3)$, and hence the number of triangles in the graph.

Specifically using the assumption that $\left|\frac{c_k}{c_3}\right| \leq h^{k-3}$ for all k, we can bound the tail terms by:

$$\begin{aligned}\left|\sum_{k=4}^{\infty} c_k\Delta^k \operatorname{tr}(\mathbf{A}^k)\right| &= \left|c_3\Delta^3\sum_{k=4}^{\infty}\frac{c_k}{c_3}\Delta^{k-3}\operatorname{tr}(\mathbf{A}^k)\right|\\ &\leq |c_3|\Delta^3\sum_{k=4}^{\infty}\frac{1}{10^{k-3}}\left(\frac{1}{n^4}\right)^{k-3}\operatorname{tr}(\mathbf{A}^k)\\ &\leq \frac{|c_3|\Delta^3}{9}, \qquad (3.5)\end{aligned}$$

where the last inequality uses the fact that $\operatorname{tr}(\mathbf{A}^k) \leq \|\mathbf{A}\|_2^{k-2}\|\mathbf{A}\|_F^2 \leq n^k \leq n^{4(k-3)}$ for all $k > 3$. Further, since $\operatorname{tr}(\mathbf{A}^0) = n$, $\operatorname{tr}(\mathbf{A}) = 0$, and $\operatorname{tr}(\mathbf{A}^2) = \|\mathbf{A}\|_F^2 \leq n^2$ we can bound the first three terms:

$$\begin{aligned}0 \leq c_0\operatorname{tr}(\mathbf{A}^0) + c_1\Delta\operatorname{tr}(\mathbf{A}) + c_2\Delta^2(\operatorname{tr}(\mathbf{A}^2)) &\leq |c_3|\Delta^3\cdot\left(\left|\frac{c_0 n}{c_3\Delta^3}\right| + \left|\frac{c_2 n^2}{c_3\Delta}\right|\right)\\ &\leq \frac{|c_3|\Delta^3}{9\epsilon_1} \qquad (3.6)\end{aligned}$$

since we set $\epsilon_1 = \frac{1}{9}\cdot\min\left\{1\ ,\ \left|\frac{c_3\Delta^3}{c_0 n}\right|\ ,\ \left|\frac{c_3\Delta}{c_2 n^2}\right|\right\}$.

Now, clearly in $O(\operatorname{nnz}(\mathbf{A}))$ time we can compute $\operatorname{tr}(\mathbf{A}^2) = \|\mathbf{A}\|_F^2$. Given spectral sum estimate $X \in [(1-\epsilon_1)\mathcal{S}_f(\mathbf{B}), (1+\epsilon_1)\mathcal{S}_f(\mathbf{B})]$ we can thus compute

$$Y \overset{\text{def}}{=} X - c_0 n - c_1\Delta\operatorname{tr}(\mathbf{A}) - c_2\Delta^2\operatorname{tr}(\mathbf{A}^2)$$

in $O(n^2)$ time. Applying (3.4), (3.5), and (3.6) we have:

$$\begin{aligned}
Y &\leq (1-\epsilon_1)\cdot\sum_{k=0}^{\infty} c_k\Delta^k \operatorname{tr}(\mathbf{A}^k)\\
&\leq c_3\Delta^3\operatorname{tr}(\mathbf{A}^3) + \frac{|c_3|\Delta^3}{9} + \epsilon_1\left(\frac{|c_3|\Delta^3}{9} + c_3\Delta^3\operatorname{tr}(\mathbf{A}^3) + \frac{|c_3|\Delta^3}{9\epsilon_1}\right)\\
&\leq c_3\Delta^3\left[\operatorname{tr}(\mathbf{A}^3)\,(1+\epsilon_1) + \frac{1}{3}\right]\\
&\leq c_3\Delta^3\left[\operatorname{tr}(\mathbf{A}^3)\left(1+\frac{1}{9}\right) + \frac{1}{3}\right].
\end{aligned}$$

We can symmetrically show that

$$Y \geq c_3\Delta^3\left[\operatorname{tr}(\mathbf{A}^3)\left(1-\frac{1}{9}\right) - \frac{1}{3}\right].$$

Thus if G contains a triangle and so $\operatorname{tr}(\mathbf{A}^3) \geq 6$, $Y \geq c_3\Delta^3\cdot\frac{15}{3}$, while if G contains no triangles and $\operatorname{tr}(\mathbf{A}^3) = 0$, $Y \leq c_3\Delta^3\cdot\frac{1}{3}$. So we can use Y to detect if G has a triangle. This gives the theorem recalling that, given X, we can compute Y in $O(\operatorname{nnz}(\mathbf{A}))$ time. □

Note that in the above reduction, if Δ is small (i.e., $\leq 2n$) then $\mathbf{B} = \mathbf{I} - \Delta\mathbf{A}$ is a very well conditioned matrix (its condition number is at most a constant). Consequently, our bounds apply even when approximately applying for example $\mathbf{B}^{-1}$ or $\mathbf{B}^{1/2}$ to high precision is inexpensive. The theorem (and the results in Section 3.2.4) suggests that the difficulty in computing spectral sums arises more from the need to measure the contribution from multiple terms precisely, than from the difficulty in manipulating $\mathbf{B}$ for the purposes of applying it to a single vector.

Also, note that the matrix $\mathbf{B}$ in this reduction is symmetrically diagonally dominant (SDD). So, even for these highly structured matrices, which admit near linear time application of $\mathbf{B}^{-1}$ [ST04] as well as approximate factorization [KS16], accurate spectral sums are difficult. We leverage this in the effective resistance hardness results of Section 3.2.4.

3.2.3 Hardness for Computing Spectral Sums

We now use Theorem 3.2.1 and Corollary 3.2.2 to show hardness results for various important spectral sums, including all those discussed in Section 3.1.1. To simplify our presentation, we focus on the case of dense matrices, showing bounds of the form

$\Omega(n^\gamma \epsilon^{-c})$. However, note that Theorem 3.2.1 and Corollary 3.2.2 also yield conditional lower bounds on the running time for sparse matrices and can be stated in terms of $\mathrm{nnz}(\mathbf{A})$. Most of the proofs in this section are direct instantiations of Corollary 3.2.2 with a specific spectral sum. Typically, the series expansion required in (3.3) is given via a simple Taylor series expansion.

Schatten p-norm for all $p \neq 2$

For any $p \in \mathbb{R}$ and $x \in (0,2)$, using the Taylor Series about 1 we can write

$$x^p = \sum_{k=0}^{\infty} c_k (1-x)^k \text{ where } c_k = \frac{\prod_{i=0}^{k-1}(p-i)}{k!}. \tag{3.7}$$

This series converges since $|c_k| \leq 1$ for $k > p$ and for $x \in (0,2)$, $(1-x) < 1$. Note that when p is a non-negative integer, only the first p terms of the expansion are nonzero. When p is non-integral, the sum is infinite. We will apply Theorem 3.2.1 and Corollary 3.2.2 slightly differently for different values of p. We first give our strongest result, which is for $p = 3$:

Corollary 3.2.3 (Schatten 3-Norm Hardness). *Assume there exists an algorithm that, for some $\delta \geq 0$, on any input $\mathbf{B} \in \mathbb{R}^{n \times n}$ returns, with probability $\geq 1-\delta$, $X \in [(1-\epsilon)\|\mathbf{B}\|_3^3, (1+\epsilon)\|\mathbf{B}\|_3^3]$ in $O(n^\gamma \epsilon^{-c})$ time. Then one can solve triangle detection on n-node graphs with success probability $\geq 1-\delta$ in $O(n^{\gamma+4c})$ time.*

Proof. For $p = 3$, in the expansion of (3.7), $c_k = 0$ for $k > 3$. So we apply Corollary 3.2.2 with $h = 0$ and hence $\Delta = 1/n$ and $\epsilon_1 = \frac{c_3 \Delta^3}{c_0 n} = \frac{1}{n^4}$. This gives the result. □

Note that for p very close to 3 a similar bound holds since $h \approx 0$. For $p = 3$, Theorem 3.1.2 gives a randomized algorithm running in $\tilde{O}(n^2/\epsilon^3)$ time and succeeding with high probability. Significant improvement to the ϵ dependence in this algorithm therefore either requires loss in the n dependence or would lead to a significant improvement in state-of-the-art triangle detection, giving an algorithm running faster than the fastest known matrix multiplication algorithm.

We next extend to all $p \neq 1, 2$. This more general result gives a bound for $p = 3$, but it is weaker than the bound given in Corollary 3.2.3.

Corollary 3.2.4 (Schatten p-Norm Hardness, $p \neq 1,2$). *For any $p > 0$, $p \neq 1, 2$, assume there exists an algorithm that, for some $\delta \geq 0$, on any input $\mathbf{B} \in \mathbb{R}^{n \times n}$ returns, with probability $\geq 1-\delta$, $X \in [(1-\epsilon)\|\mathbf{B}\|_p^p, (1+\epsilon)\|\mathbf{B}\|_p^p]$ in $O(n^\gamma \epsilon^{-c})$ time.*

Then one can solve triangle detection on n-node graphs with success probability $\geq 1-\delta$ in $O\left(n^{\gamma+13c} \cdot \frac{p^{3c}}{|\min\{p,(p-1),(p-2)\}|^c}\right)$ time.

Proof. In the expansion of (3.7) we have $\frac{c_k}{c_3} \leq p^{k-3}$ for all $k > 3$ as well as:

$$\left|\frac{c_0}{c_3}\right| = \left|\frac{1}{p(p-1)(p-2)}\right| \leq \left|\frac{1}{2\min\{p,(p-1),(p-2)\}}\right|.$$

We also have $\left|\frac{c_2}{c_3}\right| \leq \left|\frac{1}{2\min\{p,(p-1)\}}\right|$. We thus apply Corollary 3.2.2 with $\Delta = \Theta\left(\frac{1}{n^4 p}\right)$ and $\epsilon_1 = \frac{c_3\Delta^3}{c_0 n} = \Theta\left(\frac{|\min\{p,(p-1),(p-2)\}|}{n^{13}p^3}\right)$, which gives the result. □

Typically, when $p << n$, the p^{3c} term above is negligible. The $\frac{1}{|\min\{p,(p-1),(p-2)\}|^c}$ term is meaningful however. Our bound becomes weak as p approaches 2 (and meaningless when $p = 2$). This is unsurprising, since for p very close to 2, $\|\mathbf{B}\|_p^p \approx \|\mathbf{B}\|_F^2$, which can be computed exactly in $O(\text{nnz}(\mathbf{B})) = O(n^2)$ time. The bound also becomes weak for $p \approx 1$, which is natural since reduction only uses PSD $\mathbf{B}$, for which $\|\mathbf{B}\|_1 = \text{tr}(\mathbf{B})$ can be computed in $O(n)$ time. However, we can remedy this issue by working with a (non-PSD) square root of $\mathbf{B}$ that is easy to compute. Since $\|\mathbf{B}\|_1$ will correspond to the Schatten-1/2 norm of this matrix, we will be able to obtain a stronger lower bound on computing it via Corollary 3.2.4.

Corollary 3.2.5 (Schatten p-Norm Hardness, $p \approx 1$)**.** *For any p, assume there exists an algorithm that, for some $\delta \geq 0$, on any input $\mathbf{B} \in \mathbb{R}^{p\times n}$ returns, with probability $\geq 1-\delta$, $X \in [(1-\epsilon)\|\mathbf{B}\|_p^p, (1+\epsilon)\|\mathbf{B}\|_p^p]$ in $O\left(f(\text{nnz}(\mathbf{B}), n) \cdot \frac{1}{\epsilon^c}\right)$ time. Then one can solve triangle detection on n-node graphs with m edges, with success probability $\geq 1-\delta$ in*

$$O\left(f(m,n) \cdot n^{13c} \cdot \frac{p^{3c/2}}{|\min\{p/2,(p/2-1),(p/2-2)\}|^c} + m + n\right)$$

time.

Note that for $p \approx 1$, $\frac{p^{3c/2}}{|\min\{p/2,(p/2-1),(p/2-2)\}|^c}$ is just a constant. Again, the bound is naturally weak when $p \approx 2$ since $(p/2-1)$ goes to 0

Proof. Let $\mathbf{C} = I - \Delta\mathbf{A}$ as in Theorem 3.2.1. Let $\mathbf{L} = \mathbf{D} - \mathbf{A}$ be the Laplacian of G where $\mathbf{D}$ is the diagonal degree matrix. We can write $\mathbf{C} = \Delta\mathbf{L} + \widehat{\mathbf{D}}$ where $\widehat{\mathbf{D}} = \mathbf{I} - \Delta\mathbf{D}$. $\widehat{\mathbf{D}}$ is PSD since $\Delta \leq 1/n$. Letting $\mathbf{M} \in \mathbb{R}^{\binom{n}{2}\times n}$ be the vertex edge incidence matrix of $\mathbf{A}$, and $\mathbf{B} = [\Delta^{1/2}\mathbf{M}^T, \widehat{\mathbf{D}}^{1/2}]$, we have $\mathbf{B}\mathbf{B}^T = \mathbf{C}$. Thus, $\|\mathbf{B}\|_p^p = \|\mathbf{C}\|_{p/2}^{p/2}$ and so approximating this norm gives triangle detection by Corollary 3.2.4. Note that

$\text{nnz}(\mathbf{B}) = O(\text{nnz}(\mathbf{A}) + n) = O(m + n)$ and $\mathbf{B}$ can be formed in this time, giving our final runtime claim. □

For $p = 1$, Theorem 33 of [MNS+18] gives of $\tilde{O}(\epsilon^{-3}\left(mn^{1/3} + n^{3/2}\right)) = \tilde{O}(n^{2.33})$ even when $m = \Omega(n^2)$. Thus, significantly improving this ϵ dependence would either give a triangle detection algorithm running faster than the fastest known matrix multiplication algorithm, or come at a cost in the polynomials of the other parameters.

SVD Entropy

Our hardness result for SVD entropy follows as a direct consequence of Corollary 3.2.2. We simply need to expand $x \log x$ via a Taylor series.

Corollary 3.2.6 (SVD Entropy Hardness)**.** *Assume there exists an algorithm that, for some $\delta \geq 0$, on any input $\mathbf{B} \in \mathbb{R}^{n\times n}$ returns, with probability $\geq 1-\delta$, $X \in [(1-\epsilon)\sum_{i=1}^n f(\sigma_i(\mathbf{B})), (1+\epsilon)\sum_{i=1}^n f(\sigma_i(\mathbf{B}))]$ for $f(x) = x\log x$ in $O(n^\gamma \epsilon^{-c})$ time. Then one can solve triangle detection on n-node graphs with success probability $\geq 1-\delta$ in $O(n^{\gamma+6c})$ time.*

Proof. For $x \in (0, 2)$, using the Taylor Series about 1 we can write

$$x \log x = \sum_{k=0}^{\infty} c_k (x-1)^k,$$

where $c_0 = 1\log(1) = 0$, $c_1 = \log(1) + 1 = 1$, and $|c_k| = \frac{(k-2)!}{k!} \leq 1$ for $k \geq 2$. $c_k < c_3$ for all $k > 3$ and $\frac{c_0}{c_3} = 0$ while $\frac{c_2}{c_3} = \frac{1}{3}$. Applying Corollary 3.2.2 with $\Delta = \frac{1}{10n^4}$ and $\epsilon_1 = \frac{\Delta}{3n^2} = \frac{1}{30n^6}$ gives the result. □

Log Determinant

Our log determinant result also follows easily using a Taylor series expansion and then applying Corollary 3.2.2.

Corollary 3.2.7 (Log Determinant Hardness)**.** *Assume there exists an algorithm that, for some $\delta \geq 0$, on any input $\mathbf{B} \in \mathbb{R}^{n\times n}$, with probability $\geq 1-\delta$, returns $X \in [(1-\epsilon)\log(\det(\mathbf{B})), (1+\epsilon)\log(\det(\mathbf{B}))]$ in $O(n^\gamma\epsilon^{-c})$ time. Then one can solve triangle detection on n-node graphs with success probability $\geq 1-\delta$ in $O(n^{\gamma+6c})$ time.*

Proof. Using the Taylor Series about 1 we can write

$$\log x = \sum_{k=0}^{\infty} c_k (x-1)^k,$$

where $c_0 = 0$, $|c_i| = 1/i$ for $i \geq 1$. Therefore $c_k < c_3$ for all $k > 3$, $\frac{c_0}{c_3} = 0$, and $\frac{c_2}{c_3} = \frac{3}{2}$. Applying Corollary 3.2.2 with $\Delta = \frac{1}{10n^4}$ and $\epsilon_1 = \frac{\Delta}{2n^2} = \frac{1}{20n^6}$ gives the result. □

In Section 3.2.5, Lemma 3.2.13 we show that a similar result holds for computing $\det(\mathbf{B}) = \prod_{i=1}^{n} \lambda_i(\mathbf{B})$. In [BS83] it is shown that, given an arithmetic circuit for computing $\det(\mathbf{B})$, one can generate a circuit of the same size (up to a constant) that computes $\mathbf{B}^{-1}$. This also yields a circuit for matrix multiplication by a classic reduction.[6] Our results, combined with the reduction of [WW10] of Boolean matrix multiplication to triangle detection (Theorem 3.1.8), show that a sub-cubic time algorithm for the approximating $\log(\det(\mathbf{B}))$ or $\det(\mathbf{B})$ up to sufficient accuracy, yields a sub-cubic time matrix multiplication algorithm, providing a reduction based connection between determinant and matrix multiplication analogous to the circuit based result of [BS83].

Trace of Exponential

Finally, we give our bound for the trace exponential, which is again via Corollary 3.2.2 and a simple Taylor series expansion.

Corollary 3.2.8 (Trace of Exponential Hardness)**.** *Assume there exists an algorithm that, for some $\delta \geq 0$, on any input $\mathbf{B} \in \mathbb{R}^{n \times n}$, with probability $\geq 1 - \delta$, returns $X \in [(1-\epsilon)\operatorname{tr}(\exp(\mathbf{B})), (1+\epsilon)\operatorname{tr}(\exp(\mathbf{B}))]$ in $O(n^\gamma \epsilon^{-c})$ time. Then one can solve triangle detection on n-node graphs with success probability $1 - \delta$ in $O(n^{\gamma + 13c})$ time.*

Proof. Using the Taylor Series about 1 we can write

$$e^x = \sum_{k=0}^{\infty} \frac{e(x-1)^k}{k!}.$$

We have $\frac{c_0}{c_3} = 6$, $\frac{c_2}{c_3} = 3$, and for all $k \geq 3$, $c_k < c_3$. Applying Corollary 3.2.2 with $\Delta = \frac{1}{10n^4}$ and $\epsilon_1 = \frac{c_3 \Delta^3}{c_0 n} = \frac{1}{6000n^{13}}$ gives the result. □

3.2.4 Leverage Score and Effective Resistance Hardness

We now we show hardness for precisely computing all effective resistances of a graph and leverage scores of a matrix. Our main result is an easy corollary of Theorem 3.2.1, which shows how to solve triangle detection using an algorithm that precisely approximates the trace inverse of a strictly symmetric diagonally dominant (SDD) $\mathbf{B}$, i.e.,

[6] Matrix multiplication reduces to inversion since $\begin{bmatrix} \mathbf{I} & \mathbf{A} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{B} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix}^{-1} = \begin{bmatrix} \mathbf{I} & -\mathbf{A} & \mathbf{AB} \\ \mathbf{0} & \mathbf{I} & -\mathbf{B} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix}$. See [Isa08].

$\mathbf{B} = \mathbf{B}^T$ and $\mathbf{B}_{ii} > \sum_{j \neq i} \mathbf{B}_{ij}$ for all i. We will then show in Lemma 3.2.11 that approximating all the effective resistances of a certain graph is enough to approximate this trace inverse. These effective resistances correspond to the leverage scores of the graphs vertex edge incidence matrix, letting us prove hardness for leverage score computation in Corollary 3.2.12.

Corollary 3.2.9 (Trace of Inverse Hardness)**.** *Assume there exists an algorithm that, for some $\delta \geq 0$, for any strictly SDD $\mathbf{B} \in \mathbb{R}^{n \times n}$ with non-positive off-diagonal entries and m nonzero entries returns, with probability $\geq 1 - \delta$, $X \in [(1-\epsilon)\operatorname{tr}(\mathbf{B}^{-1}), (1+\epsilon)\operatorname{tr}(\mathbf{B}^{-1})]$ in $O(m^\gamma \epsilon^{-c})$ time for $\gamma \geq 2$. Then one can solve triangle detection on n-node graphs with success probability $\geq 1 - \delta$ in $O(m^{\gamma + 13c})$ time.*

Proof. For $x \in (0,2)$ we can write $\frac{1}{x} = \sum_{k=0}^{\infty} (1-x)^k$, and then apply Corollary 3.2.2 with $\Delta = \frac{1}{10n^4}$ and $\epsilon_1 = \frac{\Delta^3}{n} = \frac{1}{1000n^{13}}$. Checking that the $\mathbf{B}$ in the proof of Theorem 3.2.1 is strictly SDD with non-positive off-diagonal entries yields the result. □

Using Corollary 3.2.9 we prove hardness for precisely computing effective resistances in a graph. Recall that for a weighted undirected graph G its Laplacian is given by $\mathbf{L} = \mathbf{D} - \mathbf{A}$ where $\mathbf{D}$ is the diagonal degree matrix and $\mathbf{A}$ is the weighted adjacency matrix associated with G.

We first define the effective resistance. Throughout this section we let $\mathbf{e}_i$ denote the vector with a 1 at its i^{th} position and zeros every where else. Its length will be apparent from context. We let $\mathbf{1}$ denote the all ones vector.

Definition 3.2.10 (Effective Resistance)**.** *Given graph G with n nodes and associated Laplacian $\mathbf{L} \in \mathbb{R}^{n \times n}$, the effective resistance between vertices i and j is given by:*

$$(\mathbf{e}_i - \mathbf{e}_j)^T \mathbf{L}^+ (\mathbf{e}_i - \mathbf{e}_j)$$

where $+$ denotes the Moore-Penrose pseudoinverse.

In the following lemma we prove that computing all the effective resistances between a vertex and its neighbors in the graph can be used to compute the trace of the inverse of any strictly SDD matrix with non-positive off-diagonals and therefore doing this precisely is as hard as triangle detection via Corollary 3.2.9. Our proof is based off a standard reduction between solving strictly SDD matrices with negative off-diagonals and solving Laplacian systems.

Lemma 3.2.11 (Effective Resistance Yields Trace)**.** *Suppose we have an algorithm that, for some $\delta \geq 0$, given Laplacian $\mathbf{L} \in \mathbb{R}^{n \times n}$ with m-non-zero entries, entry*

$i \in [n]$, and error $\epsilon \in (0,1)$ computes, with probability $\geq 1-\delta$, a $1 \pm \epsilon$ approximation to the total effective resistance between i and the neighbors of i in the graph associated with $\mathbf{L}$, that is, letting $Y = \sum_{j\in[n]} (\mathbf{e}_i - \mathbf{e}_j)^T \mathbf{L}^+ (\mathbf{e}_i - \mathbf{e}_j)$, the algorithm computes

$$X \in [(1-\epsilon)Y, (1+\epsilon)Y]$$

in time $O(m^\gamma \epsilon^{-c})$. Then there is an algorithm that computes, with probability $\geq 1-\delta$, a $(1\pm\epsilon)$ approximation to the trace of the inverse of any $n\times n$ strictly SDD matrix with m non-zero entries and non-positive off-diagonals in $O(m^\gamma \epsilon^{-c})$ time. By Corollary 3.2.9 there is thus an algorithm for triangle detection on n-node graphs that succeeds with probability $\geq 1-\delta$ and runs in $O(m^{\gamma+13c})$ time.

Proof. Let $\mathbf{M} \in \mathbb{R}^{n\times n}$ be an arbitrary strictly SDD matrix with non-positive off-diagonals, i.e., $\mathbf{M} = \mathbf{M}^T$, $\mathbf{M}_{ii} > \sum_{j\neq i} |\mathbf{M}_{ij}|$, and $\mathbf{M}_{ij} \leq 0$ for all $i \neq j$. Let $\mathbf{v} \overset{\text{def}}{=} \mathbf{M}\mathbf{1}$, $\alpha \overset{\text{def}}{=} \mathbf{1}^T\mathbf{M}\mathbf{1}$, and

$$\mathbf{L} \overset{\text{def}}{=} \begin{pmatrix} \mathbf{M} & -\mathbf{v} \\ -\mathbf{v}^T & \alpha \end{pmatrix} .$$

Now, by our assumption that $\mathbf{M}$ is SDD we have that $\mathbf{v} > \mathbf{0}$ entrywise and therefore $\alpha > 0$. Therefore, the off-diagonal entries of $\mathbf{L}$ are non-positive and by construction $\mathbf{L}\mathbf{1} = \mathbf{0}$. Consequently, $\mathbf{L}$ is a $(n+1)\times(n+1)$ symmetric Laplacian matrix with $\mathrm{nnz}(\mathbf{M}) + 2n + 1$ non-zero entries.

Now, consider any $\mathbf{x} \in \mathbb{R}^n$ and $y \in \mathbb{R}$ that satisfy the following for some $i \in [n]$

$$\begin{pmatrix} \mathbf{M} & -\mathbf{v} \\ -\mathbf{v}^T & \alpha \end{pmatrix} \begin{pmatrix} \mathbf{x} \\ y \end{pmatrix} = \begin{pmatrix} \mathbf{e}_i \\ -1 \end{pmatrix} .$$

Since $\mathbf{L}$ is a symmetric Laplacian and the associated graph is connected, by construction we know that the null space of $\mathbf{L}$ is given by: $\ker(\mathbf{L}) = \mathrm{span}(\{\mathbf{1}\})$. Thus there must exist such $\mathbf{x}$ and y. Furthermore, since $\mathbf{M}$ is strictly SDD it is invertible and since $\mathbf{M}\mathbf{1} = \mathbf{v}$ we have that

$$\mathbf{x} = \mathbf{M}^{-1}\left(y \cdot \mathbf{v} + \mathbf{e}_i\right) = y \cdot \mathbf{1} + \mathbf{M}^{-1}\mathbf{e}_i$$

and consequently

$$(\mathbf{e}_i - \mathbf{e}_{n+1})^T \mathbf{L}^+ (\mathbf{e}_i - \mathbf{e}_{n+1}) = \mathbf{e}_i^T \mathbf{x} - y = \mathbf{e}_i^T \mathbf{M}^{-1} \mathbf{e}_i .$$

Thus,

$$Y = \sum_{j \in [n]} (\mathbf{e}_j - \mathbf{e}_{n+1})^T \mathbf{L}^+ (\mathbf{e}_j - \mathbf{e}_{n+1}) = \operatorname{tr}(\mathbf{M}^{-1}).$$

So an algorithm for computing $X \in [(1-\epsilon)Y, (1+\epsilon)Y]$ with probability $\geq 1-\delta$ directly gives an algorithm for computing

$$X \in [(1-\epsilon)\operatorname{tr}(\mathbf{M}^{-1}), (1+\epsilon)\operatorname{tr}(\mathbf{M}^{-1})]$$

with probability $\geq 1-\delta$, giving the result. □

Using Lemma 3.2.11, we can also show that computing leverage scores of matrix to high accuracy is also difficult. This follows from the well known fact that effective resistances in graphs and leverage scores of matrices are the same up to scaling by known quantities.

Corollary 3.2.12 (Leverage Score Hardness)**.** *Suppose we have an algorithm that, for some $\delta \geq 0$, given any $\mathbf{A} \in \mathbb{R}^{n \times d}$ can compute, with probability $1-\delta$, $\widetilde{\boldsymbol{\ell}}$ that is a $1 \pm \epsilon$ multiplicative approximation to the leverage scores of $\mathbf{A}$, i.e., for $\ell_i(\mathbf{A}) = \mathbf{a}_i^T(\mathbf{A}^T\mathbf{A})^+\mathbf{a}_i$ (Definition 2.2.4),*

$$\widetilde{\ell}_i \in [(1-\epsilon)\ell_i(\mathbf{A}), (1+\epsilon)\ell_i(\mathbf{A})] \text{ for all } i \in [n]$$

in time $O(\operatorname{nnz}(\mathbf{A})^\gamma \epsilon^{-c})$. Then there is a $O(n^{2\gamma+13c})$ time algorithm for detecting if an n-node graph contains a triangle with success probability $\geq 1-\delta$.

Proof. Let $\mathbf{L} \in \mathbb{R}^{n \times n}$ be a symmetric Laplacian. Let $E = \{\{i,j\} \subseteq [n] \times [n] : \mathbf{L}_{ij} \neq 0\}$, i.e., the set of edges in the graph associated with $\mathbf{L}$. Let $m = |E|$ and $\mathbf{B} \in \mathbb{R}^{m \times n}$ be the incidence matrix associated with $\mathbf{L}$, i.e., for all $e = \{i,j\} \in E$ we have $\mathbf{B}_{e,i} = \sqrt{-\mathbf{L}_{ij}}$ and $\mathbf{B}_{e,j} = -\sqrt{-\mathbf{L}_{ij}}$ for some canonical choice of ordering of i and j and let all other entries of $\mathbf{B} = 0$. Clearly $\operatorname{nnz}(\mathbf{B}) = \operatorname{nnz}(\mathbf{L})$ and we can form $\mathbf{B}$ in $O(\operatorname{nnz}(\mathbf{L}))$ time.

It is well known and easy to check that $\mathbf{L} = \mathbf{B}^T\mathbf{B}$. Consequently, for all $e = \{i,j\} \in E$ we have

$$\mathbf{b}_i^T(\mathbf{B}^T\mathbf{B})^+\mathbf{b}_i = (-\mathbf{L}_{ij}) \cdot (\mathbf{e}_i - \mathbf{e}_j)^T \mathbf{L}^+ (\mathbf{e}_i - \mathbf{e}_j).$$

Now if we compute $\widetilde{\boldsymbol{\ell}}$ using the assumed algorithm in $O(\operatorname{nnz}(\mathbf{L})^\gamma \epsilon^{-c}) = O(n^{2\gamma}\epsilon^{-c})$ time, then since $-\mathbf{L}_{ij}$ is non-negative, in an additional $O(\operatorname{nnz}(\mathbf{L})) = O(n^2)$ time this yields (with probability $\geq 1-\delta$) a $1 \pm \epsilon$ multiplicative approximation to the sum of effective resistances between any i and all its neighbors in the graph associated with $\mathbf{L}$. The result then follows from Lemma 3.2.11. □

3.2.5 Determinant Hardness

We now show how a variation on our techniques can be used to reduce triangle detection to accurate determinant computation. The determinant is not a spectral sum, but it is equal to the product of a matrix's eigenvalues. We will leverage this fact to show that, like the spectral sums considered in Section 3.2.3 it can be used to approximation $\operatorname{tr}(\mathbf{A}^3)$ and thus the number of triangles in a graph.

Lemma 3.2.13 (Determinant Hardness)**.** *Suppose there exists an algorithm that, for some $\delta \geq 0$, on any input $\mathbf{B} \in \mathbb{R}^{n\times n}$ returns, with probability $\geq 1-\delta$, $X \in [(1-\epsilon)\det(\mathbf{B}), (1+\epsilon)\det(\mathbf{B})]$ in $O(n^\gamma \epsilon^{-c})$ time. Then one can solve triangle detection on n-node graphs with success probability $\geq 1-\delta$ in $O(n^{\gamma+12c})$ time.*

Proof. Let $\mathbf{A} \in \mathbb{R}^{n\times n}$ be the adjacency matrix of an n-node graph G. Let $\lambda_1, ..., \lambda_n$ denote its eigenvalues. Let $\mathbf{B} = \mathbf{I} + \Delta\mathbf{A}$ for some Δ that we will set later. We have:

$$\det(\mathbf{B}) = \prod_{i=1}^{n} \lambda_i(\mathbf{B}) = \prod_{i=1}^{n}(1+\Delta\lambda_i) = \sum_{k=0}^{n}\left(\Delta^k \cdot \sum_{i_1<i_2<...<i_k} \lambda_{i_1}\lambda_{i_2}...\lambda_{i_k}\right). \tag{3.8}$$

The $k=0$ term in (3.8) is 1, and the next two are easy to compute. $\Delta\sum_{i=1}^{n}\lambda_i = \Delta\operatorname{tr}(\mathbf{A}) = 0$, and $\Delta^2\sum_{i<j}\lambda_i\lambda_j = \frac{\Delta^2}{2}\left(\sum_{i,j}\lambda_i\lambda_j - \sum_i \lambda_i^2\right) = \frac{\Delta^2}{2}\sum_i \lambda_i \operatorname{tr}(\mathbf{A}) - \frac{\Delta^2}{2}\|\mathbf{A}\|_F^2 = -\Delta^2\|\mathbf{A}\|_F^2/2$. For $k=3$ we have:

$$\begin{aligned}\Delta^3 \sum_{i<j<k}\lambda_i\lambda_j\lambda_k &= \frac{\Delta^3}{3}\left(\sum_{i<j}\lambda_i\lambda_j \operatorname{tr}(\mathbf{A}) - \sum_{i\neq j}\lambda_i^2\lambda_j\right)\\ &= 0 - \frac{\Delta^3}{3}\|\mathbf{A}\|_F^2\cdot\operatorname{tr}(\mathbf{A}) + \frac{\Delta^3}{3}\operatorname{tr}(\mathbf{A}^3)\\ &= \frac{\Delta^3}{3}\operatorname{tr}(\mathbf{A}^3).\end{aligned}$$

We will bound the $k>3$ terms by:

$$\left|\Delta^k \cdot \sum_{i_1<i_2<...<i_k}\lambda_{i_1}\lambda_{i_2}...\lambda_{i_k}\right| \leq \binom{n}{k}\Delta^k\lambda_1^k \leq (n\Delta\lambda_1)^k \leq (n^2\Delta)^k$$

since $\lambda_1 \leq n$. However, in order to obtain a tighter result, we will use stronger bounds for $k = 4, 5$. These bounds are very tedious but straightforward. Specifically:

$$\begin{aligned}
\left|\Delta^4 \sum_{i<j<k<l} \lambda_i\lambda_j\lambda_k\lambda_l\right| &= \frac{\Delta^4}{4}\left|\operatorname{tr}(\mathbf{A})\sum_{i<j<k}\lambda_i\lambda_j\lambda_k - \frac{1}{2}\sum_{i\neq j\neq k}\lambda_i^2\lambda_j\lambda_k\right| \\
&= \frac{\Delta^4}{8}\left|\operatorname{tr}(\mathbf{A})\sum_{i\neq j}\lambda_i^2\lambda_j - \sum_{i\neq j}\lambda_i^2\lambda_j^2 - \sum_{i\neq j}\lambda_i^3\lambda_j\right| \\
&= \frac{\Delta^4}{8}\left|\sum_{i\neq j}\lambda_i^2\lambda_j^2 + \operatorname{tr}(\mathbf{A})\sum_{i\neq j}\lambda_i^3 - \sum_{i}\lambda_i^4\right| \\
&= \frac{\Delta^4}{8}\left|\|\mathbf{A}\|_F^2 - 2\operatorname{tr}(\mathbf{A}^4)\right| \leq \frac{\Delta^4 n^4}{4}.
\end{aligned}$$

And similarly:

$$\begin{aligned}
\left|\Delta^5 \sum_{i<j<k<l<m} \lambda_i\lambda_j\lambda_k\lambda_l\lambda_m\right| &= \frac{\Delta^5}{30}\left|\sum_{i\neq j\neq k\neq l}\lambda_i^2\lambda_j\lambda_k\lambda_l\right| \\
&= \frac{\Delta^5}{30}\left|2\sum_{i\neq j\neq k}\lambda_i^2\lambda_j^2\lambda_k + \sum_{i\neq j\neq k}\lambda_i^3\lambda_j\lambda_k\right| \\
&= \frac{\Delta^5}{30}\left|5\sum_{i\neq j}\lambda_i^2\lambda_j^3 + \sum_{i\neq j}\lambda_i^4\lambda_j\right| \\
&= \frac{\Delta^5}{30}\left|5(\sum\lambda_i^2)(\sum\lambda_i^3) - 6\sum\lambda_i^5\right| \\
&\leq \frac{\Delta^5 n^2}{6}\operatorname{tr}(\mathbf{A}^3) + \frac{\Delta^5}{5}\lambda_1\sum\lambda_i^4 \\
&\leq \frac{\Delta^5 n^2}{6}\operatorname{tr}(\mathbf{A}^3) + \frac{\Delta^5 n^5}{5}.
\end{aligned}$$

Finally, if we set $\Delta = \frac{1}{10n^4}$ then we have:

$$\begin{aligned}
\left|\sum_{k=4}^{n}\left(\Delta^k \cdot \sum_{i_1<i_2<\ldots<i_k}\lambda_{i_1}\lambda_{i_2}\ldots\lambda_{i_k}\right)\right| &\leq \frac{\Delta^4 n^4}{4} + \frac{\Delta^5 n^5}{5} + \frac{\Delta^5 n^2}{6}\operatorname{tr}(\mathbf{A}^3) + \sum_{k=6}^{\infty}(n^2\Delta)^k \\
&\leq \Delta^3\left(\frac{1}{40} + \frac{1}{500} + \frac{1}{600}\operatorname{tr}(\mathbf{A}^3) + \left(\frac{1}{10^3} + \frac{1}{10^5} + \ldots\right)\right) \\
&\leq \frac{\Delta^3}{30} + \frac{\Delta^3}{600}\operatorname{tr}(\mathbf{A}^3).
\end{aligned}$$

We then write:

$$\det(\mathbf{B}) \leq 1 - \frac{\Delta^2\|\mathbf{A}\|_F^2}{2} + \frac{\Delta^3\operatorname{tr}(\mathbf{A}^3)}{3} + \frac{\Delta^3}{30} + \frac{\Delta^3}{600}\operatorname{tr}(\mathbf{A}^3)$$

and similarly

$$\det(\mathbf{B}) \geq 1 - \frac{\Delta^2 \|\mathbf{A}\|_F^2}{2} - \frac{\Delta^3 \operatorname{tr}(\mathbf{A}^3)}{3} - \frac{\Delta^3}{30} + \frac{\Delta^3}{600} \operatorname{tr}(\mathbf{A}^3).$$

Since $1 \leq \Delta^3 \cdot 10^3 n^{12}$ and $\frac{\Delta^2 \|\mathbf{A}\|_F^2}{2} \leq \Delta^3 \cdot 5n^6$ if we compute

$$X \in [(1 - c_1/n^{12}) \det(\mathbf{B}), (1 + c_1/n^{12}) \det(\mathbf{B})]$$

for sufficiently small constant c_1 and subtract off $\left(1 - \frac{\Delta^2 \|\mathbf{A}\|_F^2}{2}\right)$, we will be able to determine if $\operatorname{tr}(\mathbf{A}^3) > 0$ and hence detect if G has a triangle. So any algorithm approximating $\det(\mathbf{B})$ to $(1 \pm \epsilon)$ error with probability $\geq 1 - \delta$ in $O(n^\gamma \epsilon^{-c})$ time yields a triangle detection algorithm running in $O(n^{\gamma + 12c})$ time and succeeding with probability $\geq 1 - \delta$. □

3.3 Lower Bounds for Kernel Approximation

In this section we prove our conditional lower bounds for computing a relative error low-rank kernel matrix approximation satisfying Definition 3.1.4.

3.3.1 Lower Bound Approach

Our lower bounds generally follow from a two step reduction. We first reduce general rectangular matrix multiplication to Gram matrix approximation (Theorem 3.3.1).[7] We then reduce Gram matrix approximation to kernel low-rank approximation, for a broad class of kernel functions (Theorems 3.3.2 and 3.3.3). In combination, this gives a reduction from rectangular matrix multiplication to kernel low-rank approximation, and thus conditional lower bound against kernel low-rank approximation assuming hardness of speeding up general rectangular matrix multiplication.

Specifically, we show that an algorithm for computing a low-rank approximation of $\mathbf{M}\mathbf{M}^T$ for any input $\mathbf{M} \in \mathbb{R}^{n \times (d+k)}$ can be used to give a fast algorithm for multiplying any two integer matrices $\mathbf{A} \in \mathbb{Z}^{n \times d}$ and $\mathbf{C} \in \mathbb{Z}^{d \times k}$. The key idea is to set $\mathbf{M} = [\mathbf{A}, w\mathbf{C}]$ where $w \in \mathbb{R}$ is a large weight. We then have:

$$\mathbf{M}\mathbf{M}^T = \begin{bmatrix} \mathbf{A}\mathbf{A}^T & w\mathbf{A}\mathbf{C} \\ w\mathbf{C}^T\mathbf{A}^T & w^2\mathbf{C}^T\mathbf{C} \end{bmatrix}.$$

[7] Given any $\mathbf{M} \in \mathbb{R}^{n \times d}$, the associate Gram matrix is $\mathbf{M}\mathbf{M}^T$.

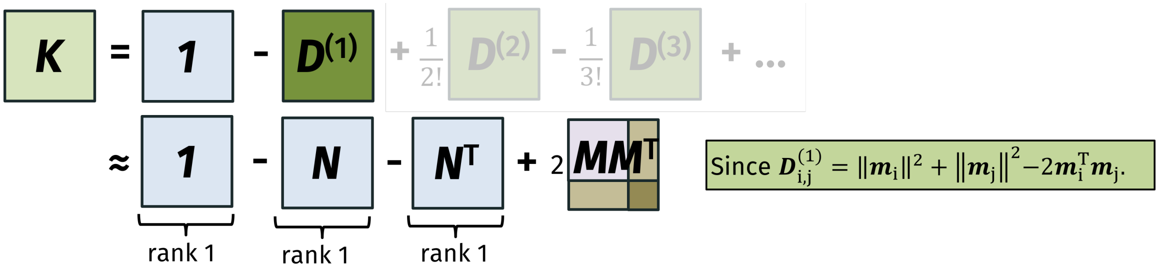

Figure 3-2: An illustration of our lower bound technique applied to the Gaussian kernel $\psi(\mathbf{m}_i, \mathbf{m}_j) = e^{-\|\mathbf{m}_i - \mathbf{m}_j\|^2}$. $\mathbf{D}^{(k)}$ denotes the powered distance matrix with $\mathbf{D}^{(k)}_{i,j} = \|\mathbf{m}_i - \mathbf{m}_j\|^{2k}$, $\mathbf{N}$ denotes the norm matrix with $\mathbf{N}_{i,j} = \|\mathbf{m}_i\|^2$, and $\mathbf{1}$ denotes the all ones matrix. Via Taylor expansion the kernel matrix $\mathbf{K}$ is well approximated by $\mathbf{1} - \mathbf{D}^{(1)}$, which equals $\mathbf{M}\mathbf{M}^T$ up to a rank-3 component $\mathbf{1} - \mathbf{N} - \mathbf{N}^T$.

Since w is very large, the $\mathbf{A}\mathbf{A}^T$ block is relatively very small, and so $\mathbf{M}\mathbf{M}^T$ is nearly rank-$2k$ – it has a 'heavy' strip of elements in its last k rows and columns. Thus, computing a relative-error rank-$2k$ approximation to $\mathbf{M}\mathbf{M}^T$ recovers all entries except those in the $\mathbf{A}\mathbf{A}^T$ block very accurately, and importantly, recovers the $w\mathbf{A}\mathbf{C}$ block and so the product $\mathbf{A}\mathbf{C}$.

We then prove that, for a broad class of kernel functions, an algorithm that, given any input $\mathbf{M} \in \mathbb{R}^{n \times (d+k)}$, computes a rank-$O(k)$ approximation of the associated kernel matrix $\mathbf{K} \in \mathbb{R}^{n \times n}$ satisfying Definition 3.1.4 can be used to obtain a close approximation to the Gram matrix $\mathbf{M}\mathbf{M}^T$. We do this by writing the kernel function $\psi(\mathbf{m}_i, \mathbf{m}_j)$ as a function of $\mathbf{m}_i^T\mathbf{m}_j$ for dot product kernels (or $\|\mathbf{m}_i - \mathbf{m}_j\|^2$ for distance kernels) and expanding this function as a power series. We show that the if input points are appropriately rescaled, the contribution of the degree-1 term $\mathbf{m}_i^T\mathbf{m}_j$ dominates, and hence our kernel matrix approximates $\mathbf{M}\mathbf{M}^T$, up to some easy to compute low-rank components. Thus, a low-rank approximation to the kernel matrix yields a low-rank approximation to $\mathbf{M}\mathbf{M}^T$, and, in turn, rectangular matrix multiplication using the reduction described above. An illustration of this general technique, as applied to the Gaussian kernel $\psi(\mathbf{m}_i, \mathbf{m}_j) = e^{-\|\mathbf{m}_i - \mathbf{m}_j\|^2}$ is shown in Figure 3-2.

3.3.2 Lower Bound for Gram Matrices

We start with our reduction from rectangular matrix multiplication to approximation of the Gram matrix $\mathbf{M}\mathbf{M}^T$.

Theorem 3.3.1 (Hardness of low-rank approximation for $\mathbf{M}\mathbf{M}^T$)**.** *Assume there is an algorithm $\mathcal{A}$ that, for some $\delta \geq 0$ and any n, d, k: for some approximation factor Δ_1,* *given any $\mathbf{M} \in \mathbb{R}^{n \times d}$, returns in $T(\mathbf{M}, k)$ time $\mathbf{N} \in \mathbb{R}^{n \times k}$ such that, with probability $\geq 1 - \delta$,*

$$\|\mathbf{M}\mathbf{M}^T - \mathbf{N}\mathbf{N}^T\|_F^2 \leq \Delta_1 \|\mathbf{M}\mathbf{M}^T - (\mathbf{M}\mathbf{M}^T)_k\|_F^2.$$

For any n, d, k and any $\mathbf{A} \in \mathbb{Z}^{n \times d}$, $\mathbf{C} \in \mathbb{Z}^{d \times k}$ each with integer entries in $[-\Delta_2, \Delta_2]$, let $\mathbf{B} = [\mathbf{A}^T, w\mathbf{C}]^T$ where $w = 3\sqrt{\Delta_1}\Delta_2^2 nd$. Then there is an algorithm that computes the product $\mathbf{AC}$ exactly with probability $\geq 1 - \delta$ in $T(\mathbf{B}, 2k) + O(nk^{\bar{\omega}-1})$ time.[8]

Proof. We can write the $(n+k) \times (n+k)$ matrix $\mathbf{B}\mathbf{B}^T$ as:

$$\mathbf{B}\mathbf{B}^T = [\mathbf{A}^T, w\mathbf{C}]^T [\mathbf{A}, w\mathbf{C}] = \begin{bmatrix} \mathbf{A}\mathbf{A}^T & w\mathbf{AC} \\ w\mathbf{C}^T\mathbf{A}^T & w^2\mathbf{C}^T\mathbf{C} \end{bmatrix}.$$

Let $\mathbf{Q} \in \mathbb{R}^{n \times 2k}$ be an orthogonal span for the columns of the $n \times 2k$ matrix:

$$\begin{bmatrix} \mathbf{0} & w\mathbf{AC} \\ \mathbf{V} & w^2\mathbf{C}^T\mathbf{C} \end{bmatrix},$$

where $\mathbf{V} \in \mathbb{R}^{k \times k}$ spans the columns of $w\mathbf{C}^T\mathbf{A}^T \in \mathbb{R}^{k \times n}$. Here and throughout $\mathbf{0}$ denotes the all zeros matrix, whose size will be apparent from context. The projection $\mathbf{Q}\mathbf{Q}^T\mathbf{B}\mathbf{B}^T$ gives the best Frobenius norm approximation to $\mathbf{B}\mathbf{B}^T$ in the span of $\mathbf{Q}$. Additionally, $\mathbf{Q}\mathbf{Q}^T\mathbf{B}\mathbf{B}^T$ has rank $2k$. So we can see that:

$$\begin{aligned} \|\mathbf{B}\mathbf{B}^T - (\mathbf{B}\mathbf{B}^T)_{2k}\|_F^2 &\leq \|\mathbf{B}\mathbf{B}^T - \mathbf{Q}\mathbf{Q}^T\mathbf{B}\mathbf{B}^T\|_F^2 \\ &\leq \left\| \begin{bmatrix} \mathbf{A}\mathbf{A}^T & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \right\|_F^2 \\ &\leq \Delta_2^4 n^2 d^2, \end{aligned} \tag{3.9}$$

where the last bound follows since each entry of $\mathbf{A}$ is bounded in magnitude by Δ_2 by assumption and so each entry of $\mathbf{A}\mathbf{A}^T$ is bounded by $d\Delta_2^2$.

Let $\mathbf{N}$ be the matrix returned by running $\mathcal{A}$ on $\mathbf{B}$ with rank $2k$. By assumption, with probability $\geq 1 - \delta$, $\mathbf{N}$ achieves approximation bound of $\|\mathbf{B}\mathbf{B}^T - \mathbf{N}\mathbf{N}^T\|_F^2 \leq \Delta_1 \|\mathbf{B}\mathbf{B}^T - (\mathbf{B}\mathbf{B}^T)_{2k}\|_F^2$. This gives, for all i, j:

$$(\mathbf{B}\mathbf{B}^T - \mathbf{N}\mathbf{N}^T)_{i,j}^2 \leq \|\mathbf{B}\mathbf{B}^T - \mathbf{N}\mathbf{N}^T\|_F^2 \leq \Delta_1 \Delta_2^4 n^2 d^2,$$

[8] Here $\bar{\omega}$ is the lowest known exponent of fast matrix multiplication. See Section 1.3 for a formal definition.

where the last inequality is from (3.9). This gives

$$|\mathbf{B}\mathbf{B}^T - \mathbf{N}\mathbf{N}^T|_{i,j} \leq \sqrt{\Delta_1}\Delta_2^2 nd. \tag{3.10}$$

Since $\mathbf{A}$ and $\mathbf{C}$ have integer entries, each entry in the submatrix $w\mathbf{A}\mathbf{C}$ of $\mathbf{B}\mathbf{B}^T$ is an integer multiple of $w = 3\sqrt{\Delta_1}\Delta_2^2 nd$. Since by (3.10) $(\mathbf{N}\mathbf{N}^T)_{i,j}$ approximates this entry to error $\sqrt{\Delta_1}\Delta_2^2 nd$, by simply rounding $(\mathbf{N}\mathbf{N}^T)_{i,j}$ to the nearest multiple of w, we obtain the entry exactly. Thus, given $\mathbf{N}$, we can exactly recover $\mathbf{A}\mathbf{C}$ in $O(nk^{\bar{\omega}-1})$ time by computing the $n \times k$ submatrix of $\mathbf{N}\mathbf{N}^T$ corresponding to $\mathbf{A}\mathbf{C}$ in $\mathbf{B}\mathbf{B}^T$. This approach succeeds if $\|\mathbf{B}\mathbf{B}^T - \mathbf{N}\mathbf{N}^T\|_F^2 \leq \Delta_1 \|\mathbf{B}\mathbf{B}^T - (\mathbf{B}\mathbf{B}^T)_{2k}\|_F^2$, which by assumption occurs with probability $\geq 1 - \delta$. □

In the case of the linear kernel $\psi(\mathbf{m}_i, \mathbf{m}_j) = \mathbf{m}_i^T \mathbf{m}_j$, Theorem 3.3.1 immediately gives our main result (full stated in Theorem 3.1.9), which shows that rectangular matrix multiplication can be reduced to kernel low-rank approximation.

Theorem 3.1.9 (Hardness for low-rank kernel approximation – linear kernel)**.** *Consider the linear kernel $\psi(\mathbf{m}_i, \mathbf{m}_j) = \mathbf{m}_i^T \mathbf{m}_j$. Assume there is an algorithm that, for some $\delta \geq 0$ and any n, d, k: for some approximation factor Δ, given $\mathbf{M} \in \mathbb{R}^{n \times d}$ with associated kernel matrix $\mathbf{K} = \{\psi(\mathbf{m}_i, \mathbf{m}_j)\} = \mathbf{M}\mathbf{M}^T$, returns in $o(\mathrm{nnz}(\mathbf{M})k + nk^p)$ time, for $p \geq 2$, $\mathbf{N} \in \mathbb{R}^{n \times k}$ satisfying, with probability $\geq 1 - \delta$,*

$$\|\mathbf{K} - \mathbf{N}\mathbf{N}^T\|_F^2 \leq \Delta \|\mathbf{K} - \mathbf{K}_k\|_F^2.$$

Then for any n, d, k, there is an $o(\mathrm{nnz}(\mathbf{A})k + nk^p)$ time algorithm that, given arbitrary $\mathbf{A} \in \mathbb{Z}^{n \times d}$, $\mathbf{C} \in \mathbb{Z}^{d \times k}$, returns their product $\mathbf{A}\mathbf{C}$ with probability $\geq 1 - \delta$.

Proof of Theorem 3.1.9 – Linear Kernel. We apply Theorem 3.3.1 after noting that for $\mathbf{B} = [\mathbf{A}^T, w\mathbf{C}]^T$, $\mathrm{nnz}(\mathbf{B}) \leq \mathrm{nnz}(\mathbf{A}) + nk$ and so the runtime given by theorem is:

$$\begin{aligned} T(\mathbf{B}, 2k) + O(nk^{\bar{\omega}-1}) &= o(\mathrm{nnz}(\mathbf{B}) \cdot 2k + n(2k)^p) + O(nk^{\bar{\omega}-1}) \\ &= o(\mathrm{nnz}(\mathbf{A})k + nk^p). \end{aligned}$$

□

As discussed, any algorithm for computing $\mathbf{A}\mathbf{C}$ in $o(\mathrm{nnz}(\mathbf{A})k)$ time for general $\mathbf{A} \in \mathbb{Z}^{n \times d}$, $\mathbf{C} \in \mathbb{Z}^{d \times k}$ would be a major breakthrough in fast matrix multiplication. Thus Theorem 3.1.9 shows that giving an algorithm for computing a low-rank approximation of $\mathbf{M}\mathbf{M}^T$ running in $o(\mathrm{nnz}(\mathbf{M})k)$ time is unlikely.

We show in Section 3.3.5 that there is an algorithm that nearly matches the conditional lower bound of Theorem 3.1.9 for $\psi(\mathbf{m}_i, \mathbf{m}_j) = \mathbf{m}_i^T \mathbf{m}_j$ for $\Delta = (1+\epsilon)$ for any $\epsilon > 0$. For constant ϵ, this algorithm slightly improves the runtime of the general PSD low-rank approximation algorithm, Algorithm 1, presented in Chapter 2 and analyzed in Theorem 2.5.1. In Section 3.3.6 we show that even just outputting an orthonormal matrix $\mathbf{Z} \in \mathbb{R}^{n \times k}$ such that $\tilde{\mathbf{K}} = \mathbf{Z}\mathbf{Z}^T\mathbf{M}\mathbf{M}^T$ is a relative-error low-rank approximation of $\mathbf{M}\mathbf{M}^T$, but not computing a factorization of $\tilde{\mathbf{K}}$ itself, is enough to give fast multiplication of integer matrices.

3.3.3 Lower Bound for Dot Product Kernels

We now extend Theorem 3.3.1 to general dot product kernels – where $\psi(\mathbf{a}_i, \mathbf{a}_j) = f(\mathbf{a}_i^T \mathbf{a}_j)$ for some function f. This class includes, for example, the polynomial kernel, $\psi(\mathbf{a}_i, \mathbf{a}_j) = (c + \mathbf{a}_i^T \mathbf{a}_j)^q$, for some c, q. Our proof argues that, if the function $f(x)$ has a power series expansion in which the coefficients of the higher order terms are not too large compared to the coefficient of x, then for appropriate weights w_1, w_2, the kernel matrix corresponding to $\mathbf{B} = [w_1 \mathbf{A}^T, w_2 \mathbf{C}]^T$ is dominated by the first order component, which corresponds to $\mathbf{B}\mathbf{B}^T$. As shown in Section 3.3.2 above, an approximation to this product can be used to exactly compute $\mathbf{AC}$.

Theorem 3.3.2 (Hardness of low-rank approximation for dot product kernels)**.** *Consider any kernel $\psi : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^+$ with $\psi(\mathbf{a}_i, \mathbf{a}_j) = f(\mathbf{a}_i^T \mathbf{a}_j)$ for some function f that can be expanded as $f(x) = \sum_{q=0}^{\infty} c_q x^q$ with $c_1 \neq 0$ and $|c_q/c_1| \leq G^{q-1}$ and for all $q \geq 2$ and some G.*

Assume there is an algorithm $\mathcal{A}$ that, for some $\delta \geq 0$ and any n, d, k: for some approximation factor Δ_1, given $\mathbf{M} \in \mathbb{R}^{n \times d}$ with kernel matrix $\mathbf{K} = \{\psi(\mathbf{m}_i, \mathbf{m}_j)\}$, returns in $T(\mathbf{M}, k)$ time $\mathbf{N} \in \mathbb{R}^{n \times k}$ satisfying, with probability $\geq 1 - \delta$,

$$\|\mathbf{K} - \mathbf{N}\mathbf{N}^T\|_F^2 \leq \Delta_1 \|\mathbf{K} - \mathbf{K}_k\|.$$

For any $\mathbf{A} \in \mathbb{Z}^{n \times d}$, $\mathbf{C} \in \mathbb{Z}^{d \times k}$ with integer entries in $[-\Delta_2, \Delta_2]$, let $\mathbf{B} = [w_1 \mathbf{A}^T, w_2 \mathbf{C}]^T$ with $w_1 = \frac{w_2}{12\sqrt{\Delta_1}\Delta_2^2 nd}$, $w_2 = \frac{1}{4\sqrt{Gd}\Delta_2}$. Then there is an algorithm that computes the product $\mathbf{AC}$ exactly with probability $\geq 1 - \delta$ in $T(\mathbf{B}, 2k+1) + O(nk^{\bar{\omega}-1})$ time.

Proof. Using our decomposition of $\psi(\cdot, \cdot)$, we can write the kernel matrix for $\mathbf{B}$ and

ψ as:

$$\mathbf{K} = c_0 \begin{bmatrix} \mathbf{1} & \mathbf{1} \\ \mathbf{1} & \mathbf{1} \end{bmatrix} + c_1 \begin{bmatrix} w_1^2 \mathbf{A}\mathbf{A}^T & w_1 w_2 \mathbf{A}\mathbf{C} \\ w_1 w_2 \mathbf{C}^T \mathbf{A}^T & w_2^2 \mathbf{C}^T \mathbf{C} \end{bmatrix} + c_2 \mathbf{K}^{(2)} + c_3 \mathbf{K}^{(3)} + ... \tag{3.11}$$

where $\mathbf{K}^{(q)}_{i,j} = (\mathbf{b}_i^T \mathbf{b}_j)^q$ and $\mathbf{1}$ denotes the all ones matrix of appropriate size. The key idea is to show that the contribution of the $\mathbf{K}^{(q)}$ terms is small, and so any relative-error rank-$(2k+1)$ approximation to $\mathbf{K}$ must recover an approximation to $\mathbf{B}\mathbf{B}^T$, and thus the product $\mathbf{AC}$ as in Theorem 3.3.1.

By our setting of $w_2 = \frac{1}{4\sqrt{G}d\Delta_2}$, the fact that $w_1 < w_2$, and our bound on the entries of $\mathbf{A}$ and $\mathbf{C}$, we have for all i, j,

$$|\mathbf{b}_i^T \mathbf{b}_j| \leq w_2^2 d \Delta_2^2 < \frac{1}{16G}.$$

Thus, for any i, j, using that $|c_q/c_1| \leq G^{q-1}$:

$$\begin{aligned} \left| \sum_{q=2}^{\infty} c_q \mathbf{K}^{(q)}_{i,j} \right| &\leq c_1 |\mathbf{b}_i^T \mathbf{b}_j| \cdot \left| \sum_{q=2}^{\infty} G^{q-1} |\mathbf{b}_i^T \mathbf{b}_j|^{q-1} \right| \\ &\leq c_1 |\mathbf{b}_i^T \mathbf{b}_j| \sum_{q=2}^{\infty} \frac{G^{q-1}}{(16G)^{q-1}} \\ &\leq \frac{1}{12} c_1 |\mathbf{b}_i^T \mathbf{b}_j|. \end{aligned} \tag{3.12}$$

Let $\bar{\mathbf{K}}$ be the matrix $\left(\mathbf{K} - c_0 \begin{bmatrix} \mathbf{1} & \mathbf{1} \\ \mathbf{1} & \mathbf{1} \end{bmatrix} \right)$, with its top right $n \times n$ block set to 0. $\bar{\mathbf{K}}$ has just its last k columns and rows non-zero, so has rank $\leq 2k$. Let $\mathbf{Q} \in \mathbb{R}^{n \times 2k+1}$ be an orthogonal span for the columns $\bar{\mathbf{K}}$ along with the all ones vector of length n. Let $\mathbf{N}$ be the result of running $\mathcal{A}$ on $\mathbf{B}$ with rank $2k+1$. Then, with probability $\geq 1 - \delta$:

$$\begin{aligned} \|\mathbf{K} - \mathbf{N}\mathbf{N}^T\|_F^2 \leq \Delta_1 \|\mathbf{K} - \mathbf{K}_{2k+1}\|_F^2 &\leq \Delta_1 \|\mathbf{K} - \mathbf{Q}\mathbf{Q}^T \mathbf{K}\|_F^2 \\ &\leq \Delta_1 \left\| \begin{bmatrix} (c_1 w_1^2 \mathbf{A}\mathbf{A}^T + c_2 \hat{\mathbf{K}}^{(2)} + ...) & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \right\|_F^2 \end{aligned} \tag{3.13}$$

where $\hat{\mathbf{K}}^{(q)}$ denotes the top left $n \times n$ submatrix of $\mathbf{K}^{(q)}$. By our bound on the entries of $\mathbf{A}$ and (3.12):

$$\left| \left(c_1 w_1^2 \mathbf{A}\mathbf{A}^T + c_2 \hat{\mathbf{K}}^{(2)} + c_3 \hat{\mathbf{K}}^{(3)} + ... \right)_{i,j} \right| \leq \frac{13}{12} \left| \left(c_1 w_1^2 \mathbf{A}\mathbf{A}^T \right)_{i,j} \right| \leq 2 c_1 w_1^2 d \Delta_2^2.$$

Plugging back into (3.13) and using $w_1 = \frac{w_2}{12\sqrt{\Delta_1}\Delta_2^2 nd}$, this gives for any i, j:

$$\begin{aligned}(\mathbf{K} - \mathbf{N}\mathbf{N}^T)_{i,j} \leq \|\mathbf{K} - \mathbf{N}\mathbf{N}^T\|_F &\leq \sqrt{\Delta_1 n^2} \cdot 2c_1 w_1^2 d\Delta_2^2 \\ &\leq \frac{\sqrt{\Delta_1}n \cdot 2c_1 d\Delta_2^2}{12\sqrt{\Delta_1}\Delta_2^2 nd} \cdot w_1 w_2 \\ &\leq \frac{w_1 w_2 c_1}{6}. \end{aligned} \tag{3.14}$$

Since $\mathbf{A}$ and $\mathbf{C}$ have integer entries, each entry of $c_1 w_1 w_2 \mathbf{AC}$ is an integer multiple of $c_1 w_1 w_2$. By the decomposition of (3.11) and the bound of (3.12), if we subtract c_0 from the corresponding entry of $\mathbf{K}$ and round it to the nearest multiple of $c_1 w_1 w_2$, we will recover the entry of $\mathbf{AC}$. By the bound of (3.14), we can likewise round the corresponding entry of $\mathbf{N}\mathbf{N}^T$. Computing all nk of these entries given $\mathbf{N}$ takes time $O(nk^{\bar{\omega}-1})$, giving the theorem. □

Theorem 3.3.2 lets us bound the time to compute a low-rank kernel approximation for any kernel function expressible as a reasonable power expansion of $\mathbf{a}_i^T \mathbf{a}_j$. As a straightforward example, it gives the conditional lower bound for the polynomial kernel of any degree stated in Theorem 3.1.9.

Theorem 3.1.9 (Hardness for low-rank kernel approximation – polynomial kernel)**.** *Consider the polynomial kernel $\psi(\mathbf{m}_i, \mathbf{m}_j) = (c + \mathbf{m}_i^T \mathbf{m}_j)^q$. Assume there is an algorithm that, for some $\delta \geq 0$ and any n, d, k: for some approximation factor Δ, given $\mathbf{M} \in \mathbb{R}^{n \times d}$ with associated kernel matrix $\mathbf{K} = \{\psi(\mathbf{m}_i, \mathbf{m}_j)\} = \mathbf{M}\mathbf{M}^T$, returns in $o(\text{nnz}(\mathbf{M})k + nk^p)$ time, for $p \geq 2$, $\mathbf{N} \in \mathbb{R}^{n \times k}$ satisfying, with probability $\geq 1 - \delta$,*

$$\|\mathbf{K} - \mathbf{N}\mathbf{N}^T\|_F^2 \leq \Delta \|\mathbf{K} - \mathbf{K}_k\|_F^2.$$

Then for any n, d, k, there is an $o(\text{nnz}(\mathbf{A})k + nk^p)$ time algorithm that, given arbitrary $\mathbf{A} \in \mathbb{Z}^{n \times d}$, $\mathbf{C} \in \mathbb{Z}^{d \times k}$, returns their product $\mathbf{AC}$ with probability $\geq 1 - \delta$.

Proof of Theorem 3.1.9 – Polynomial Kernel. We can write $\psi(\mathbf{m}_i, \mathbf{m}_j) = (c + \mathbf{m}_i^T \mathbf{m}_j)^q$ as $f(\mathbf{m}_i^T \mathbf{m}_j)$ where $f(x) = \sum_{j=0}^q c_j x^j$ with $c_j = c^{q-j}\binom{q}{j}$. Thus $c_1 \neq 0$ and $|c_j / c_1| \leq G^{j-1}$ for $G = (q/c)$, allowing us to apply Theorem 3.3.2. Finally note that $\text{nnz}(\mathbf{B}) \leq \text{nnz}(\mathbf{A}) + nk$ and so the runtime given by the theorem is:

$$\begin{aligned} T(\mathbf{B}, 2k+1) + O(nk^{\bar{\omega}-1}) &= o(\text{nnz}(\mathbf{B}) \cdot (2k+1) + n(2k+1)^p) + O(nk^{\bar{\omega}-1}) \\ &= o(\text{nnz}(\mathbf{A})k + nk^p), \end{aligned}$$

which yields the result. □

3.3.4 Lower Bound for Distance Kernels

Finally, we extend Theorem 3.3.2 to handle kernels like the Gaussian kernel whose value depends on the squared distance $\|\mathbf{a}_i - \mathbf{a}_j\|^2$ rather than just the dot product $\mathbf{a}_i^T \mathbf{a}_j$. Our proof is similar to that of Theorem 3.3.2. The key idea is to write $\mathbf{K}$ as a polynomial in the *distance matrix* $\mathbf{D}$ with $\mathbf{D}_{i,j} = \|\mathbf{b}_i - \mathbf{b}_j\|_2^2$. Since $\|\mathbf{b}_i - \mathbf{b}_j\|_2^2 = \|\mathbf{b}_i\|_2^2 + \|\mathbf{b}_j\|_2^2 - 2\mathbf{b}_i^T\mathbf{b}_j$, $\mathbf{D}$ can be written as $-2\mathbf{B}\mathbf{B}^T$ plus a rank-2 component. By setting w_1, w_2 sufficiently small, as in the proof of Theorem 3.3.2, we ensure that the higher powers of $\mathbf{D}$ are negligible, and thus that our low-rank approximation must accurately recover the submatrix of $\mathbf{B}\mathbf{B}^T$ corresponding to $\mathbf{AC}$.

Theorem 3.3.3 (Hardness of low-rank approximation for distance kernels)**.** *Consider any kernel function $\psi : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}^+$ with $\psi(\mathbf{a}_i, \mathbf{a}_j) = f(\|\mathbf{a}_i - \mathbf{a}_j\|^2)$ for some function f that can be expanded as $f(x) = \sum_{q=0}^{\infty} c_q x^q$ with $c_1 \neq 0$ and $|c_q/c_1| \leq G^{q-1}$ and for all $q \geq 2$ and some $G \geq 1$.*

Assume there is an algorithm $\mathcal{A}$ that, for some $\delta \geq 0$ and any n, d, k: for some approximation factor Δ_1, given input $\mathbf{M} \in \mathbb{R}^{n \times d}$ with kernel matrix $\mathbf{K} = \{\psi(\mathbf{m}_i, \mathbf{m}_j)\}$, returns in $T(\mathbf{M}, k)$ time $\mathbf{N} \in \mathbb{R}^{n \times k}$ satisfying, with probability $\geq 1 - \delta$,

$$\|\mathbf{K} - \mathbf{N}\mathbf{N}^T\|_F^2 \leq \Delta_1 \|\mathbf{K} - \mathbf{K}_k\|.$$

For any $\mathbf{A} \in \mathbb{Z}^{n \times d}$, $\mathbf{C} \in \mathbb{Z}^{d \times k}$ with integer entries in $[-\Delta_2, \Delta_2]$, let $\mathbf{B} = [w_1\mathbf{A}^T, w_2\mathbf{C}]^T$ with $w_1 = \frac{w_2}{36\sqrt{\Delta_1}\Delta_2^2 nd}$, $w_2 = \frac{1}{(16Gd^2\Delta_2^4)(36\sqrt{\Delta_1}\Delta_2^2 nd)}$. Then there is an algorithm that computes the product $\mathbf{AC}$ exactly with probability $\geq 1 - \delta$ in $T(\mathbf{B}, 2k+3) + O(nk^{\bar{\omega}-1})$ time.

Proof. Define the distance matrix $\mathbf{D} \in \mathbb{R}^{n+k \times n+k}$ with $\mathbf{D}_{i,j} = \|\mathbf{b}_i - \mathbf{b}_j\|^2$. Using the fact that $\|\mathbf{b}_i - \mathbf{b}_j\|^2 = \|\mathbf{b}_i\|^2 + \|\mathbf{b}_i\|^2 - 2\mathbf{b}_i^T\mathbf{b}_j$ we have:

$$\mathbf{D} = \mathbf{E} + \mathbf{E}^T - 2\mathbf{B}\mathbf{B}^T,$$

where $\mathbf{E}$ is a rank-1 matrix with all rows equal to $[\|\mathbf{b}_1\|_2^2, ..., \|\mathbf{b}_{n+k}\|_2^2]$. We can thus write the kernel matrix for $\mathbf{B}$ and ψ as:

$$\mathbf{K} = c_0 \begin{bmatrix} \mathbf{1} & \mathbf{1} \\ \mathbf{1} & \mathbf{1} \end{bmatrix} + c_1(\mathbf{E} + \mathbf{E}^T) - 2c_1 \begin{bmatrix} w_1^2 \mathbf{A}\mathbf{A}^T & w_1 w_2 \mathbf{AC} \\ w_1 w_2 \mathbf{C}^T\mathbf{A}^T & w_2^2 \mathbf{C}^T\mathbf{C} \end{bmatrix} + c_2\mathbf{D}^{(2)} + c_3\mathbf{D}^{(3)} + ... \tag{3.15}$$

where $\mathbf{D}_{i,j}^{(q)} = \|\mathbf{b}_i - \mathbf{b}_j\|^{2q}$. Let $\bar{\mathbf{K}}$ be $\mathbf{K} - c_0 \cdot \mathbf{1} - c_1(\mathbf{E} + \mathbf{E}^T)$, with its top $n \times n$

block set to 0. $\bar{\mathbf{K}}$ has rank at most $2k$ and if we set $\mathbf{Q} \in \mathbb{R}^{n \times 2k+3}$ to be a matrix with columns spanning the columns of $\bar{\mathbf{K}}$, the all ones vector, $\mathbf{E}$ and $\mathbf{E}^T$, then letting $\mathbf{N}$ be the result of running $\mathcal{A}$ on $\mathbf{B}$ with rank $2k+3$, with probability $\geq 1-\delta$:

$$\begin{aligned}\|\mathbf{K} - \mathbf{N}\mathbf{N}^T\|_F^2 &\leq \Delta_1 \|\mathbf{K} - \mathbf{Q}\mathbf{Q}^T\mathbf{K}\|_F^2 \\ &\leq \Delta_1 \left\| \begin{bmatrix} -2c_1 w_1^2 \mathbf{A}\mathbf{A}^T + c_2 \hat{\mathbf{D}}^{(2)} + \ldots & 0 \\ 0 & 0 \end{bmatrix} \right\|_F^2, \end{aligned} \tag{3.16}$$

where $\hat{\mathbf{D}}^{(q)}$ denotes the top left $n \times n$ submatrix of $\mathbf{D}^{(q)}$.

By our bounds on the entries of $\mathbf{A}$ and $\mathbf{C}$, for $i, j \leq n$, $\|\mathbf{b}_i - \mathbf{b}_j\|^2 \leq 4d\Delta_2^2 w_1^2$ and by our setting of w_1, w_2, plugging into (3.16) we have for all i, j:

$$\begin{aligned}|(\mathbf{K} - \mathbf{N}\mathbf{N}^T)_{i,j}| &\leq \|\mathbf{K} - \mathbf{N}\mathbf{N}^T\|_F & (3.17)\\ &\leq \sqrt{\Delta_1} n \left(2c_1 d \Delta_2^2 w_1^2 + \sum_{q=2}^{\infty} c_q (4d\Delta_2^2 w_1^2)^q \right) \\ &\leq \sqrt{\Delta_1} n c_1 d \Delta_2^2 w_1^2 \left(2 + \sum_{q=2}^{\infty} (4Gd\Delta_2^2 w_1^2)^{q-1} \right) & \text{(Since } |c_q/c_1| \leq G^{q-1}) \\ &\leq 3\sqrt{\Delta_1} n c_1 d \Delta_2^2 w_1^2 \\ &\leq \frac{w_1 w_2 c_1}{12}, & (3.18)\end{aligned}$$

where the second to last bound follows from the fact that $w_1 < w_2$ and w_2 is set small enough so $(4Gd\Delta_2^2) \cdot w_2^2 \ll 1/2$ so the series converges to a sum < 1. Additionally, for $i \leq n$ and $j \leq k$ (i.e., considering the entries of $\mathbf{K}$ corresponding to $\mathbf{AC}$) we have:

$$\mathbf{K}_{i,n+j} = c_0 + c_1(\mathbf{E} + \mathbf{E}^T)_{i,n+j} - 2c_1 w_1 w_2 (\mathbf{AC})_{i,j} + \sum_{q=2}^{\infty} c_q \mathbf{D}_{i,n+j}^{(q)}.$$

This last sum can be bounded by:

$$\begin{aligned}
\left|\sum_{q=2}^{\infty} c_q \mathbf{D}^{(q)}_{i,n+j}\right| &\le c_1 \sum_{q=2}^{\infty} G^{q-1}(4\Delta_2^2 d w_2^2)^q && \text{(By assumption } |c_q/c_1| \le G^{q-1}\text{)}\\
&\le c_1 w_1 w_2 \sum_{q=2}^{\infty} G^{q-1} w_2^{2(q-1)} \frac{w_2}{w_1} \left(4\Delta_2^2 d\right)^q\\
&\le c_1 w_1 w_2 \sum_{q=2}^{\infty} G^{q-1} w_2^{2q-3} \left(4\Delta_2^2 d\right)^q && \text{(Using } \tfrac{w_2}{w_1} \le \tfrac{1}{w_2}.\text{)}\\
&\le \frac{c_1 w_1 w_2}{3}. && \text{(Using } w_2 \le \tfrac{1/4}{16G\Delta_2^4 d^2} \text{ so the series converges.)}
\end{aligned}$$

If we set $v = \mathbf{N}\mathbf{N}^T_{i,n+j} - c_0 - c_1(\mathbf{E} + \mathbf{E}^T)_{i,n+j}$ we thus have combining with (3.17) for $i \le n$, $j \le k$

$$|v + 2c_1 w_1 w_2 (\mathbf{A}\mathbf{C})_{i,j}| \le \frac{5c_1 w_1 w_2}{12}$$

and so we can compute $(\mathbf{A}\mathbf{C})_{i,j}$ exactly by rounding v to the nearest integer multiple of $c_1 w_1 w_2$. This gives the theorem since we can compute the required entries of $\mathbf{N}\mathbf{N}^T$ and $\mathbf{E}$ in $O(nk^{\bar{\omega}-1})$ time. □

As a simple example, Theorem 3.3.3 gives the conditional lower bound for the Gaussian kernel stated in Theorem 3.1.9. The Gaussian kernel is one of the most widely used kernels in machine learning, appearing commonly in the kernel method literature and serving as the building block of Gaussian process methods [Ras04]. Its approximation has also been studied, for example, in the numerical linear algebra literature [GS91].

Theorem 3.1.9 (Hardness for low-rank kernel approximation – Gaussian kernel)**.** *Consider the Gaussian kernel $\psi(\mathbf{m}_i, \mathbf{m}_j) = e^{-\|\mathbf{m}_i - \mathbf{m}_j\|^2/\sigma}$. Assume there is an algorithm that, for some $\delta \ge 0$ and any n, d, k: for some approximation factor Δ, given $\mathbf{M} \in \mathbb{R}^{n \times d}$ with associated kernel matrix $\mathbf{K} = \{\psi(\mathbf{m}_i, \mathbf{m}_j)\} = \mathbf{M}\mathbf{M}^T$, returns in $o(\mathrm{nnz}(\mathbf{M})k + nk^p)$ time, for $p \ge 2$, $\mathbf{N} \in \mathbb{R}^{n \times k}$ satisfying with probability $\ge 1 - \delta$:*

$$\|\mathbf{K} - \mathbf{N}\mathbf{N}^T\|_F^2 \le \Delta \|\mathbf{K} - \mathbf{K}_k\|_F^2.$$

Then for any n, d, k, there is an $o(\mathrm{nnz}(\mathbf{A})k + nk^p)$ time algorithm that, given arbitrary $\mathbf{A} \in \mathbb{Z}^{n \times d}$, $\mathbf{C} \in \mathbb{Z}^{d \times k}$, returns their product $\mathbf{A}\mathbf{C}$ with probability $\ge 1 - \delta$.

Proof of Theorem 3.1.9 – Gaussian Kernel. $\psi(m_i, m_j)$ can be written as $f(\|\mathbf{m}_i -$

$\mathbf{m}_j\|^2)$ where

$$f(x) = e^{-x/\sigma} = \sum_{q=0}^{\infty} \frac{(-1/\sigma)^q}{q!} x^q.$$

Thus $c_1 \neq 0$ and $|c_q/c_1| \leq G^{q-1}$ for $G = 1/\sigma$. Applying Theorem 3.3.3 and bounding $\text{nnz}(\mathbf{B}) \leq \text{nnz}(\mathbf{A}) + nk$, gives the result since the runtime given by the theorem is:

$$\begin{aligned} T(\mathbf{B}, 2k+3) + O(nk^{\bar{\omega}-1}) &= o(\text{nnz}(\mathbf{B}) \cdot (2k+3) + n(2k+3)^p) + O(nk^{\bar{\omega}-1}) \\ &= o(\text{nnz}(\mathbf{A})k + nk^p). \end{aligned}$$

□

3.3.5 Fast Low-Rank Approximation of $\mathbf{AA}^T$

In this section we give an algorithm that matches the conditional lower bound of Theorem 3.1.9 for the linear kernel. This algorithm improves on the general PSD low-rank approximation algorithm of [MW17b] (Algorithm 1 presented in Chapter 2 and analyzed in Theorem 2.5.1) by improving logarithmic factors and ϵ dependencies. We present it here so that it can be directly compared to our lower bound.

Theorem 3.3.4. *There is an algorithm that, given $\mathbf{A} \in \mathbb{R}^{n \times d}$, $k \in \mathbb{Z}^{\geq 1}$, and $\epsilon > 0$, computes $\mathbf{N} \in \mathbb{R}^{n \times k}$ in $O(\text{nnz}(\mathbf{A})k) + n \cdot \text{poly}(k/\epsilon)$ time such that probability $\geq 99/100$:*

$$\|\mathbf{AA}^T - \mathbf{NN}^T\|_F^2 \leq (1+\epsilon)\|\mathbf{AA}^T - (\mathbf{AA}^T)_k\|_F^2.$$

Proof. It is known (see Lemma 11 of [CW17b]) that there exists a distribution over random matrices $\mathbf{R}, \mathbf{S} \in \mathbb{R}^{n \times O(k/\epsilon)}$ that can be applied to $\mathbf{A}$ in $O(\text{nnz}(\mathbf{A})) + n \cdot \text{poly}(k/\epsilon)$ time such that with probability $\geq 199/200$, setting

$$\mathbf{Y}^* = \underset{\mathbf{Y} \in O(k/\epsilon) \times O(k/\epsilon) \text{ with rank } k}{\arg\min} \|\mathbf{AA}^T\mathbf{RYS}^T\mathbf{AA}^T - \mathbf{AA}^T\|_F^2$$

we have:

$$\|\mathbf{AA}^T\mathbf{RY}^*\mathbf{S}^T\mathbf{AA}^T - \mathbf{AA}^T\|_F^2 \leq (1+\epsilon)\|\mathbf{AA}^T - (\mathbf{AA}^T)_k\|_F^2.$$

We can solve for an approximately optimal $\tilde{\mathbf{Y}}$ by further sketching our problem on the left and right (similar to the technique used in Lemma 15 of [CW17b]). Specifically, if we let $\mathbf{T}_L, \mathbf{T}_R \in \mathbb{R}^{n \times \text{poly}(k/\epsilon)}$ be drawn from the Count Sketch distribution, we can solve:

$$\tilde{\mathbf{Y}} = \underset{\mathbf{Y}\in O(k/\epsilon)\times O(k/\epsilon) \text{ with rank } k}{\arg\min} \|\mathbf{T}_L^T\mathbf{A}\mathbf{A}^T\mathbf{R}\mathbf{Y}\mathbf{S}^T\mathbf{A}\mathbf{A}^T\mathbf{T}_R - \mathbf{T}_L^T\mathbf{A}\mathbf{A}^T\mathbf{T}_R\|_F^2$$

and are guaranteed that with probability $\geq 99/100$,

$$\|\mathbf{A}\mathbf{A}^T\mathbf{R}\tilde{\mathbf{Y}}\mathbf{S}^T\mathbf{A}\mathbf{A}^T - \mathbf{A}\mathbf{A}^T\|_F^2 \leq (1+2\epsilon)\|\mathbf{A}\mathbf{A}^T - (\mathbf{A}\mathbf{A}^T)_k\|_F^2. \tag{3.19}$$

Computing $\tilde{\mathbf{Y}}$ requires forming $\mathbf{T}_L^T\mathbf{A}$, $\mathbf{A}^T\mathbf{R}$, $\mathbf{S}^T\mathbf{A}$, and $\mathbf{A}^T\mathbf{T}_R$ and then multiplying the appropriate matrices together. This takes $O(\text{nnz}(\mathbf{A})) + n\,\text{poly}(k/\epsilon)$ time. Once $\mathbf{T}_L^T\mathbf{A}\mathbf{A}^T\mathbf{R}$, $\mathbf{S}^T\mathbf{A}\mathbf{A}^T\mathbf{T}_R$ and $\mathbf{T}_L^T\mathbf{A}\mathbf{A}^T\mathbf{T}_R$ have been formed we can solve for $\tilde{\mathbf{Y}}$ in $\text{poly}(k/\epsilon)$ time using the formula of [FT07].

Finally, since $\tilde{\mathbf{Y}}$ is rank-k we can factor $\tilde{\mathbf{Y}} = \mathbf{V}\mathbf{V}^T$ for $\mathbf{V} \in \mathbb{R}^{O(k/\epsilon)\times k}$ using the SVD. We can then compute $\mathbf{N}_1 = \mathbf{A}\mathbf{A}^T\mathbf{R}\mathbf{V} \in \mathbb{R}^{n\times k}$ and $\mathbf{N}_3 = \mathbf{A}\mathbf{A}^T\mathbf{S}\mathbf{V} \in \mathbb{R}^{n\times k}$ which satisfy $\|\mathbf{A}\mathbf{A}^T - \mathbf{N}_1\mathbf{N}_2^T\|_F^2 \leq (1+2\epsilon)\|\mathbf{A}\mathbf{A}^T - (\mathbf{A}\mathbf{A}^T)_k\|_F^2$ with probability $\geq 99/100$ by (3.19).

$\mathbf{N}_1$ and $\mathbf{N}_2$ both require $O(\text{nnz}(\mathbf{A})k)+n\cdot\text{poly}(k/\epsilon)$ time to compute. The theorem follows from adjusting constants on ϵ and noting that we can symmetrize $\mathbf{N}_1\mathbf{N}_2^T$ to form $\mathbf{N}\mathbf{N}^T$ if desired in $n \cdot \text{poly}(k/\epsilon)$ time.

□

3.3.6 Hardness of Outputting a Low-Rank Subspace

Theorems 3.3.1 and 3.1.9 show a conditional lower bound on outputting a relative-error low-rank approximation to $\mathbf{M}\mathbf{M}^T$ for any $\mathbf{M} \in \mathbb{R}^{n\times d}$. Here we show that this hardness extends to the possibly easier problem of just outputting a low-rank span that contains a relative-error low-rank approximation. This result extends analogously to the other kernel lower bounds discussed in Section 3.3.

Theorem 3.3.5 (Hardness of low-rank span for $\mathbf{M}\mathbf{M}^T$)**.** *Assume there is an algorithm $\mathcal{A}$ that, for some $\delta \geq 0$ and any n, d, k: for some approximation factor Δ_1, given any $\mathbf{M} \in \mathbb{R}^{n\times d}$ returns in $T(\mathbf{M}, k)$ time orthonormal $\mathbf{Z} \in \mathbb{R}^{n\times k}$ such that with probability $\geq 1-\delta$,*

$$\|\mathbf{M}\mathbf{M}^T - \mathbf{Z}\mathbf{Z}^T\mathbf{M}\mathbf{M}^T\|_F^2 \leq \Delta_1\|\mathbf{M}\mathbf{M}^T - (\mathbf{M}\mathbf{M}^T)_k\|_F^2.$$

For any $\mathbf{A} \in \mathbb{Z}^{n\times d}$, $\mathbf{C} \in \mathbb{Z}^{d\times k}$ with integer entries in $[-\Delta_2, \Delta_2]$, let $\mathbf{B} = [\mathbf{A}^T, w\mathbf{C}]^T$ where $w = 3\sqrt{\Delta_1}\Delta_2^2 nd$. Then, for any fixed c, there is an algorithm that computes the product $\mathbf{A}\mathbf{C}$ exactly with probability $\geq 1-\delta-1/k^c$ in $T(\mathbf{B}, 2k) + \tilde{O}((n+d)k^{\bar{\omega}-1})$

time.

Proof. $\mathbf{Z}\mathbf{Z}^T\mathbf{M}\mathbf{M}^T$ is the projection of $\mathbf{M}\mathbf{M}^T$ onto the column span of $\mathbf{Z}$. This projection can be performed approximately using standard leverage score sampling techniques, similar to those discussed in Chapter 2. Let $\mathbf{S} \in \mathbb{R}^{s\times n}$ be a sampling matrix sampling rows of $\mathbf{Z}$ by its row norms (its leverage scores since it is orthonormal) where $s = c_1(k \log k)$ or some constant c_1. Let $\mathbf{R} \in \mathbb{R}^{n\times k}$ have its bottom $k \times k$ submatrix be an identity matrix and its top $(n-k) \times k$ submatrix be $\mathbf{0}$.

Letting $\mathbf{X}^* = \arg\min_{\mathbf{X}\in k\times k} \|\mathbf{Z}\mathbf{X}^T - \mathbf{M}\mathbf{M}^T\mathbf{R}\|_F^2$ and $\mathbf{X} = \arg\min_{\mathbf{X}\in k\times k} \|\mathbf{S}\mathbf{Z}\mathbf{X}^T - \mathbf{S}\mathbf{M}\mathbf{M}^T\mathbf{R}\|_F^2$ we have by a well known leverage score approximate regression result that, for any fixed c, if c_1 (the constant in the sample size s) is large enough, with probability $\geq 1 - 1/k^c$:

$$\begin{aligned}
\|\mathbf{Z}\mathbf{X}^T - \mathbf{M}\mathbf{M}^T\mathbf{R}\|_F^2 &= O(1) \cdot \|\mathbf{Z}(\mathbf{X}^*)^T - \mathbf{M}\mathbf{M}^T\mathbf{R}\|_F^2 \\
&= O(1) \cdot \|\mathbf{Z}\mathbf{Z}^T\mathbf{M}\mathbf{M}^T\mathbf{R} - \mathbf{M}\mathbf{M}^T\mathbf{R}\|_F^2 \\
&= O(\Delta_1)\|\mathbf{M}\mathbf{M}^T - (\mathbf{M}\mathbf{M}^T)_k\|_F^2.
\end{aligned}$$

Further, computing $\mathbf{X}$ takes $\tilde{O}(dk^{\bar{\omega}-1})$ time to compute the $O(k \log k) \times k$ submatrix $\mathbf{S}\mathbf{M}\mathbf{M}^T\mathbf{R}$ as well as $\tilde{O}(k^{\bar{\omega}}) = \tilde{O}(nk^{\bar{\omega}-1})$ to perform the regression. This gives the result via Theorem 3.3.1 since computing $\mathbf{Z}$ with rank-$2k$ $\mathbf{Z}\mathbf{X}^T$ gives a low-rank approximation of $\mathbf{M}\mathbf{M}^T$ with error $O(\Delta_1)\|\mathbf{M}\mathbf{M}^T - (\mathbf{M}\mathbf{M}^T)_{2k}\|_F^2$ *measured on the last* k *columns of* $\mathbf{M}$. Small error on these columns is all that is needed to recover $\mathbf{A}\mathbf{C}$ accurately (see the proof of Theorem 3.3.1). □

3.4 Discussion and Future Work

The work in this section represents a few preliminary steps in understanding complexity of linear algebraic primitives like spectral summarization (Section 3.2) and low-rank approximation (Section 3.3). There are many interesting open directions in developing lower bounds for linear algebraic problems, which we discuss below.

3.4.1 Connecting Matrix Multiplication to Other Problems

In Section 3.2 we show that general square matrix multiplication can be reduced to highly accurate approximation of a number of spectral sum problems, like the Schatten p-norms, the log determinant, and the trace inverse. Our reductions are all by way of a reduction of [WW10] from matrix multiplication to triangle detection

(Theorem 3.1.8). Surprisingly, outside of this reduction, very few reductions from matrix multiplication to other problems are known. In particular, many problems which seem to require matrix multiplication time to solve are not known to be as hard as matrix multiplication. We mention a few examples here that we believe are important to explore, both from the lower bound and algorithmic directions.

- For general $\mathbf{A} \in \mathbb{R}^{n\times n}$ there are no known algorithms that compute even a constant factor multiplicative approximation to $\det(\mathbf{A})$ or $\lambda_{\min}(\mathbf{A})$ in $o(n^\omega)$ time (i.e., without performing a full eigendecomposition). However, it is unclear if matrix multiplication can be reduced to either of these problems. In Theorem 3.2.13 we reduce triangle detection to determinant approximation up to $(1 \pm 1/\operatorname{poly}(n))$ accuracy. Via Theorem 3.1.8 this further gives a reduction from matrix multiplication to determinant approximation up to $(1 \pm 1/\operatorname{poly}(n))$ accuracy. However, nothing is known if just a constant factor approximation to the determinant is required.

- One of the most fundamental problems in linear algebra is linear system solving: given positive semidefinite $\mathbf{A} \in \mathbb{R}^{n\times n}$ and $\mathbf{x} \in \mathbb{R}^n$, approximate $\mathbf{A}^{-1}\mathbf{x}$. If $\mathbf{A}$ has condition number κ, $\mathbf{A}^{-1}\mathbf{x}$ can be approximated in $\tilde{O}(\operatorname{nnz}(\mathbf{A}) \cdot \sqrt{\kappa})$ time [Saa03]. If $\mathbf{A}$ is structured (e.g. tridiagonal [GE95] or a graph Laplacian [ST04]), $\tilde{O}(\operatorname{nnz}(\mathbf{A}))$ or $\tilde{O}(n^2)$ time system solvers also exist. However, for general $\mathbf{A}$, no known $o(n^\omega)$ time algorithm for approximating $\mathbf{A}^{-1}\mathbf{x}$ to relatively high accuracy exists. At the same time it is not known how the complexity of solving PSD linear systems relates to the complexity of matrix multiplication. For example, is it possible to show a reduction which proves that, if all positive semidefinite systems can be solved in $\tilde{O}(n^2)$ time, then matrix multiplication can be performed in $\tilde{O}(n^2)$ time?

3.4.2 Understanding the Role of Randomness

As we have seen, randomized algorithms have led to significant progress in fast approximation of linear algebraic problems. Typically, randomization goes hand-in-hand with approximation, yielding algorithms that achieve $(1+\epsilon)$ error guarantees in some metric at the cost of $\operatorname{poly}(1/\epsilon)$ factors in their runtimes. Our lower bounds in Section 3.2 demonstrate that, for a wide variety of spectral sum problems, this coarse approximation is required to obtain fast algorithms. Algorithms with fine enough approximation can be used to give fast algorithms for general matrix multiplication.

However, our lower bounds don't address the question of if randomness is also necessary to obtain fast algorithms for approximating these spectral sums. Do fast, deterministic, spectral sum approximation algorithms exist? Below we outline a few additional open questions that we feel are important in understanding the role of randomness in fast linear algebra.

- The top eigenvalue, or relatedly the spectral norm of any matrix can be approximated up to error $(1 \pm \epsilon)$ using the power method [Saa11] or the Lanczos method, which runs in $\tilde{O}(\mathrm{nnz}(\mathbf{A} \cdot \sqrt{1/\epsilon})$ time [KW92]. These iterative algorithms however, require a random initialization vector. No algorithm is known that can compute even a constant factor approximation to the spectral norm deterministically in $o(n^\omega)$ time. Proving a lower bound or providing a new deterministic algorithm to help close this wide gap would be a very interesting result. As an intermediate step, it may also be worth studying Las Vegas algorithms for spectral norm approximation, of which none running in $o(n^\omega)$ time are known.

- Like spectral norm approximation, all known fast algorithms for low-rank approximation are randomized. For example, as discussed in Chapter 2, a near optimal k-rank approximation of general $\mathbf{A} \in \mathbb{R}^{n \times d}$ (Problem 2.1.1) can be computed in $O(\mathrm{nnz}(\mathbf{A}) + (n+d)\,\mathrm{poly}(k/\epsilon))$ using random sketching methods [CW13]. However, the fastest known algorithm for computing such an approximation deterministically, Liberty's Frequent Directions algorithm, runs in $O(\mathrm{nnz}(\mathbf{A}) \cdot k/\epsilon)$ time [GLPW16, LACBL16]. It would be very interesting to prove a lower bound showing that input sparsity runtimes (i.e. $O(\mathrm{nnz}(\mathbf{A})$ plus lower order terms) can only be achieved with the use of randomness.

Chapter 4

Ant-Inspired Density Estimation

In this chapter we study the problem of distributed population density estimation in ant colonies from the viewpoint of randomized algorithms. Many ant species employ distributed population density estimation in applications ranging from quorum sensing [Pra05], to task allocation [Gor99], to appraisal of enemy colony strength [Ada90]. It has been shown that ants estimate density by tracking encounter rates – the higher the population density, the more often the ants bump into each other [Pra05, GPT93].

We study distributed density estimation from a theoretical perspective. We prove that a group of anonymous agents randomly walking on a grid are able to estimate their density within a small multiplicative error in few steps by measuring their rates of encounter with other agents. Despite dependencies inherent in the fact that nearby agents may collide repeatedly (and, worse, cannot recognize when this happens), our bound nearly matches what would be required to estimate density by independently sampling grid locations.

From a biological perspective, our work helps shed light on how ants and other social insects can obtain relatively accurate density estimates via encounter rates. From a technical perspective, our analysis provides new tools for understanding complex dependencies in the collision probabilities of multiple random walks. We bound the strength of these dependencies using *local mixing properties* of the underlying graph. Our results extend beyond the grid to more general graphs and we discuss applications to size estimation for social networks and density estimation in robot swarms.

This chapter covers work originally published in [MSL17]. In related work, we study how approximate density estimation may be used as a subroutine in the house-hunting process of *Temnothorax* ants [RML17], specifically considering its use in the algorithms proposed in [GMRL15].

Remark: The results presented in this chapter were developed jointly with Nancy Lynch and Hsin-Hao Su. Hsin-Hao originally wrote the details of some of the proofs for density estimation on more general graphs (Section 4.4), although their presentation has been modified significantly in this thesis. We would like to thank Yury Polyanskiy for pointing our a bug in our original proofs, which assumed independence of collision counts between agents. This bug has been corrected in this writeup.

4.1 Background and Introduction to Results

The ability to sense local population density is an important tool used by many ant species. When a colony of *Temnothorax* ants must relocate to a new nest, scouts search for potential nest sites, assess their quality, and recruit other scouts to high quality locations. A high enough density of scouts at a potential new nest (a *quorum threshold*) triggers those ants to decide on the site and transport the rest of the colony there [Pra05]. When neighboring colonies of *Azteca* ants compete for territory, a high relative density of a colony's ants in a contested area will cause those ants to attack enemies in the area, while a low relative density will cause the colony to retreat [Ada90]. Varying densities of harvester ants successfully performing certain tasks such as foraging or brood care can trigger other ants to switch tasks, maintaining proper worker allocation in the colony [Gor99, SHG06].

It has been shown that ants estimate density in a distributed manner, by measuring encounter rates [Pra05, GPT93]. As ants randomly walk around an area, if they bump into a larger number of other ants, this indicates a higher population density. By tracking encounters with specific types of ants, for example, successful foragers or enemies, ants can estimate more specific densities. This strategy allows each ant to obtain an accurate density estimate and requires very little communication – ants must simply detect when they collide and do not need to perform any higher level data aggregation.

4.1.1 Density Estimation on the Grid

We study distributed density estimation from a theoretical perspective. We model a colony of ants as a set of anonymous agents randomly placed on a two-dimensional grid. Computation proceeds in rounds, with each agent stepping in a random direction in each round. A *collision* occurs when two agents reach the same position in the same round and encounter rate is measured as the number of collisions an agent is involved in during a sequence of rounds, divided by the number of rounds. Aside from collision detection, the agents have no other means of communication.

The intuition that encounter rate tracks density is clear. It is easy to show that, for a set of randomly walking agents, the *expected* encounter rate measured by each agent is exactly the density d – the number of agents divided by the grid size (see Corollary 4.3.3). However, it is unclear if the encounter rate actually gives a good density estimate – that is, if the estimate is close to its expectation with high probability.

Consider agents positioned not on the grid, but on a complete graph. In each round, each agent steps to a uniformly random position and, in expectation, the number of other agents it collides with in this step is d. Since each agent chooses its new location uniformly at random in each step, collisions are essentially *independent* between rounds. The agents are effectively taking independent Bernoulli samples with success probability d, and by a standard Chernoff bound, within $O\left(\frac{\log(1/\delta)}{d\epsilon^2}\right)$ rounds each obtains a $(1 \pm \epsilon)$ multiplicative approximation to d with probability $1 - \delta$.

On the grid graph, the picture is significantly more complex. If two agents are initially located near each other, they are more likely to collide via random walking. After a first collision, due to their proximity, they are likely to collide repeatedly in future rounds. Since the agents are anonymous, they cannot recognize repeat collisions, and even if they could, it is unclear that it would help. On average, compared to the complete graph, agents collide with fewer individuals and collide multiple times with those individuals that they do encounter, making encounter rates a less reliable estimate of population density.

Mathematically speaking, on a graph with a *fast mixing time* [Lov93], like the complete graph, each agent's location is only weakly correlated with its previous locations. This ensures that collisions are also weakly correlated between rounds and encounter rate serves as a very accurate estimate of density. The grid graph on the other hand is *slow mixing* – agent positions and hence collisions are highly correlated between rounds, lowering the accuracy of encounter-rate-based estimation.

4.1.2 Our Contributions

Surprisingly, despite the high correlation between collisions, we show that encounter-rate-based density estimation on the grid is nearly as accurate as on the complete graph. After just $O\left(\frac{\log(1/\delta)\cdot[\log\log(1/\delta)+\log(1/d\epsilon)]^2}{d\epsilon^2}\right)$ rounds, each agent's encounter rate is a $(1 \pm \epsilon)$ approximation to d with probability $1 - \delta$ (Theorem 4.3.1). This matches performance on the complete graph up to a $[\log\log(1/\delta) + \log(1/d\epsilon)]^2$ factor.

Technically, to bound accuracy on the grid, we obtain moment bounds on the number of times that two randomly walking agents collide over a set of rounds (Lemma 4.3.11). These bounds also apply to the number of equalizations (returns to origin) of a single walk. While *expected* random walk hitting times, return times, and collision rates are well studied for many graphs, including grid graphs [Lov93, ES09, KMTS16], higher moment bounds and high probability results are much less common.

Our moment bounds show that, while the grid graph is slow mixing, it has strong *local mixing*. That is, random walks tend to spread quickly over a local area and not repeatedly cover the same nodes, making random-walk-based density estimation accurate. Significant work has focused on showing that random walk sampling is nearly as good as independent sampling for fast mixing expander graphs [Gil98, CLLM12]. To the best of our knowledge, we are the first to extend this type of analysis to slowly mixing graphs, showing that strong local mixing is sufficient in many applications.

The key to the local mixing property of the grid is an upper bound on the probability that two random walks starting from the same position re-collide (or that a single random walk equalizes) after a certain number of steps (Lemma 4.3.4). We show that re-collision probability bounds imply collision moment bounds on general graphs, and apply this technique to extend our results to d-dimensional grids, regular expanders, and hypercubes. We discuss applications of our bounds to the task of estimating the size of a social network using random walks [KLSC14], obtaining improvements over prior work for networks with relatively slow global mixing times but strong local mixing. We also discuss connections to density estimation by robot swarms and random-walk-based sensor network sampling [AB04, LB07].

4.1.3 Road Map

In Section 4.2 we overview our theoretical model for distributed density estimation on the grid. In Section 4.3 we give our main technical results on random-walk-based density estimation. In Section **??** we show how to extend our bounds to a number of graphs other than the grid. In Section 4.5, as a baseline for our results, we analyze a simple algorithm for density estimation when agents are not restricted to moving by random walk. In Section 4.6 we discuss applications of our results to social network size estimation and robot swarm algorithms. In Section 4.7 we conclude and discuss interesting open questions and directions for future work.

4.2 Theoretical Model for Density Estimation

We begin by laying out our simple theoretical model of ant colony behavior and defining the density estimation problem within this model.

4.2.1 Computational Model

We consider a set of agents populating a two-dimensional torus with A nodes (dimensions $\sqrt{A} \times \sqrt{A}$). At each time step, each agent has an associated ordered pair *position*, which gives its coordinates on the torus. We assume that A is large – larger than the area agents traverse over the runtimes of our algorithms. We believe the torus model successfully captures the dynamics of density estimation on a surface, while avoiding complicating factors of boundary behavior on a finite grid.

Initially each agent is placed independently at a uniform random node in the torus. Computation proceeds in discrete, synchronous rounds. In each round, an agent may either remain in its current location or step to any of its four neighboring grid squares. Formally, it updates the ordered pair *position* by adding a step chosen from $\{(0, 1), (0, -1), (1, 0), (-1, 0), (0, 0)\}$.

A *randomly walking agent* chooses its step uniformly at random from $\{(0, 1), (0, -1), (1, 0), (-1, 0)\}$ in each round. Of course, in reality ants do not move via pure random walk. However, there is evidence that nevertheless, encounter rates are well predicted by a random walk model [BFKN18]. At the same time, there is evidence that in some cases, encounter rates are actually lower than predicted by such a model [GPT93, NTD05]. Overall, we feel that our model sufficiently captures the highly random movement of ants while remaining tractable to analysis and applicable to ant-inspired random-walk-based algorithms (Section 4.6). Extending our work to more realistic models of ant movement would be an interesting next direction. See Section 4.7.1 for a more detailed discussion of this direction.

Aside from the ability to move in each round, agents can sense the number of agents other than themselves at their position at the *end of each round*, formally through the function *count*(*position*). We say that two agents *collide in round* r if they have the same position at the end of the round. Outside of collision counting, agents have no means of communication. They are anonymous (cannot uniquely identify each other) and execute identical density estimation routines. A basic illustration of our model is depicted in Figure 4-1.

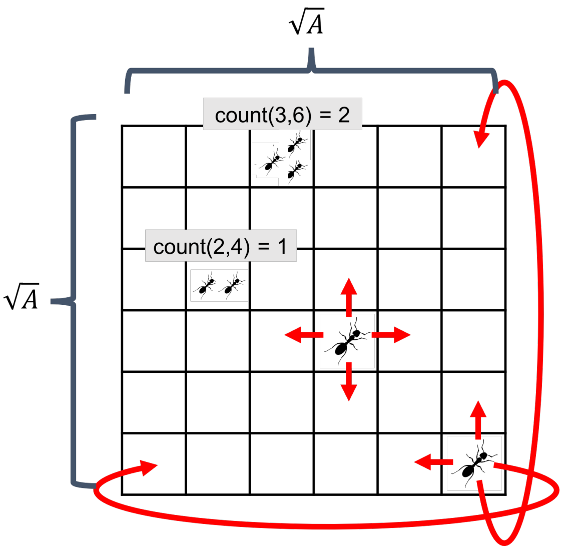

Figure 4-1: A basic illustration of our computational model. Each agent (ant) may move to an adjacent position on the two-dimensional torus in each round (illustrated by the red arrows). A collision occurs when two or more agents are located at the same position. The agents detect collisions through the *count*(*position*) function which returns the *number of other agents* at their current position. In this illustration, *position* is given as the (x, y) position with the bottom left corner corresponding to $(1, 1)$. However, the precise convention used is unimportant.

4.2.2 The Density Estimation Problem

Let $(n+1)$ be the number of agents and define population density as $d \stackrel{\text{def}}{=} n/A$. Each agent's goal is to estimate d to $(1 \pm \epsilon)$ accuracy with probability at least $1 - \delta$ for $\epsilon, \delta \in (0, 1)$ – that is, to return an estimate $\tilde{d}$ with $\mathbb{P}\left[\tilde{d} \in [(1-\epsilon)d, (1+\epsilon)d]\right] \geq 1-\delta$. As a technicality, with $n+1$ agents we define $d = n/A$ instead of $d = (n+1)/A$. In the natural case, when n is large, the distinction is unimportant. Since our analysis always takes the perspective of one agent, this convention ensures that there are n other agents with which this agent may interact, and thus all summations over expected collision counts and other quantities are over n variables rather than $n - 1$. Additionally, in the case there is a single agent on the grid, this convention allows the agent to return density estimate 0. If the density were instead defined to be $1/A$ in this case, estimation would be impossible in our model, since the agent has no way of sensing its location and thus no way of estimating the size of the torus, A.

Local vs. Global Density

The problem described above requires estimating the *global population density*. We assume that agents are initially distributed uniformly at random on the torus, which is critical for fast global density estimation. When agents are uniformly distributed, the local density in a small radius around their starting position reflects the global density with good probability. Thus, they are able to obtain a good estimate of this density using local measurements, and without traversing a large fraction of the torus.

Of course, in nature, ants are not typically uniformly distributed in the nest or surrounding areas. Additionally, they are often interested in estimating *local population densities* – e.g., around a nest entrance when estimating the number of successful foragers for task allocation [Gor99]. We view our work as a first step towards a theoretical understanding of density estimation and focus on the global density for simplicity. Removing our assumption of uniformly distributed agents, formally defining the problem of local density estimation, and understanding how ants can solve this problem are important directions for future work.

4.3 Random-Walk-Based Density Estimation

As discussed, the challenge in analyzing random-walk-based density estimation on the torus arises from correlations between collisions of nearby agents. If we do not restrict agents to random walking, and instead allow each agent to take an arbitrary step in each round, they can avoid collision correlations by splitting into 'stationary' and 'mobile' groups and counting collisions only between members of different groups. This allows them to essentially independently sample collisions with other agents to estimate density. This method is simple to analyze (see Section 4.5), but it is not 'natural' in a biological sense or useful for the applications of Section 4.6. Further, independent sampling is unnecessary! Algorithm 1 describes a simple random-walk-based approach that gives a nearly matching bound.

Algorithm 1 Random-Walk-Based Density Estimation

```
Each agent independently executes:
  $c := 0$
  for $r = 1, ..., t$ do
      $step := rand\{(0,1), (0,-1), (1,0), (-1,0)\}$
      $position := position + step$
      $c := c + count(position)$                ▷ Update collision count.
  end for
  return $\tilde{d} = \frac{c}{t}$
```

4.3.1 Random-Walk-Based Density Estimation Analysis

Our main theoretical result follows; its proof appears at the end of Section 4.3.6, after a number of preliminary lemmas. Throughout our analysis, we take the viewpoint of a single agent executing Algorithm 1, which we sometimes call *agent a*.

Theorem 4.3.1 (Random Walk Sampling Accuracy Bound). *After running for t rounds, assuming $t \leq A$, an agent executing Algorithm 1 returns $\tilde{d}$ such that, for any $\delta > 0$, with probability $\geq 1 - \delta$,*

$$\tilde{d} \in [(1-\epsilon)d, (1+\epsilon)d] \text{ for } \epsilon \leq c_1 \cdot \sqrt{\frac{\log(1/\delta)}{td}} \cdot \log(2t),$$

where c_1 is some fixed constant. This implies that, for any $\epsilon, \delta \in (0,1)$ if

$$A \geq t \geq \frac{c_2 \log(1/\delta) \cdot [\log\log(1/\delta) + \log(1/d\epsilon)]^2}{d\epsilon^2},$$

where c_2 is some fixed constant, $\tilde{d} \in [(1-\epsilon)d, (1+\epsilon)d]$ with probability $\geq 1 - \delta$.

Theorem 4.3.1 focuses on the density estimate of a single agent executing Algorithm 1. However, we note that if we set $\delta = \frac{\delta'}{n}$, then by a union bound, all n agents will have $\tilde{d} \in [(1-\epsilon)d, (1+\epsilon)d]$ with probability δ'. The required running time t will depend just logarithmically on δ' and n.

4.3.2 Decomposition of Collision Count into Independent Random Variables

We decompose the collision count c maintained by an agent executing Algorithm 1 as the sum of collisions with different agents over different rounds. Specifically, assign arbitrary ids $1, 2, ..., n$ to the n other agents and let $c_j(r)$ equal 1 if the agent collides with agent j in round r, and 0 otherwise. Let $c_j = \sum_{r=1}^{t} c_j(r)$ be the total number of collisions with agent j. We have $c = \sum_{j=1}^{n} c_j$. Note that $c_1, ..., c_n$ are identically distributed random variables.

The main challenge in proving the accuracy of Algorithm 1 is in handling the strong correlations between collisions in successive rounds – i.e., between the random variables $c_j(1), ..., c_j(t)$ for each j. Across agents, the collision counts $c_1, ..., c_n$ may also be correlated. However, conditioned on the random walk taken by agent a (the agent whose viewpoint we take), $c_1, ..., c_n$ are independent, since they depend only on the independent random walks of different agents. Thus, the results in this section will typically bound collision probabilities and expectations conditioned on agent a's path, which we denote by $\mathcal{W}$. $\mathcal{W}$ is a random variable, consisting of a sequence of t positions. In Section 4.3.6 we will remove this conditioning, showing that Algorithm 1 yields an accurate density estimate, regardless of agent a's path, and thereby proving Theorem 4.3.1.

4.3.3 Correctness of Encounter Rate in Expectation

Lemma 4.3.2 (Unbiased Estimator). *Let $\mathcal{W}$ be the t-step random walk that an agent executing Algorithm 1 takes. The output $\tilde{d}$ of that agent satisfies: $\mathbb{E}[\tilde{d}|\mathcal{W}] = d$.*

Proof. By linearity of expectation, $\mathbb{E}[c|\mathcal{W}] = \sum_{j=1}^{n} \sum_{r=1}^{t} \mathbb{E}[c_j(r)|\mathcal{W}]$. Conditioned on $\mathcal{W}$, the position of the agent is fixed in round r. Since each other agent is initially at a uniform random location and after any number of steps, is still at uniform random location, for all j, r, $\mathbb{E}[c_j(r)|\mathcal{W}] = 1/A$. Thus, $\mathbb{E}[c|\mathcal{W}] = nt/A = dt$ and $\mathbb{E}[\tilde{d}|\mathcal{W}] = \mathbb{E}[c|\mathcal{W}]]/t = d$. □

By the law of iterated expectation, $\mathbb{E}[\tilde{d}] = \mathbb{E}[\mathbb{E}[\tilde{d}|\mathcal{W}]]$ and so Lemma 4.3.2 gives:

Corollary 4.3.3. $\mathbb{E}[\tilde{d}] = d$.

We note that the torus is bipartite, and hence two agents initially located an odd number of steps away from each other will never meet via random walking. However, this fact does not change the expectation of $\tilde{d}$ computed above and in fact does not affect any of our following proofs. In future work, it may be interesting to consider a model in which each agent performs a *lazy random walk*, remaining at the same position with some probability in each round. Such a model would allow all agents to eventually meet with some probability. We discuss this direction in more detail in Section 4.7.

With Lemma 4.3.2 and Corollary 4.3.3 in place, it remains to show that the encounter rate is close to its expectation with high probability and so provides a good estimate of density. In order to do this, we must bound the strength of correlations between collisions of nearby agents in successive rounds, which can decrease the accuracy of the encounter-rate-based estimate.

4.3.4 A Re-collision Probability Bound

The key to bounding collision correlations is bounding the probability of a re-collision between two randomly walking agents in round $r+m$, assuming a collision in round r, which we do in Lemma 4.3.4 below.[1] Each c_j is the sum of highly correlated random variables $c_j(1), ..., c_j(t)$. Due to the slow mixing of the grid, if two agents collide at round r, they are much more likely to collide in successive rounds. However, by bounding this re-collision probability, we are able to give strong moment bounds for the distribution of each c_j. We bound not only its variance, but all higher moments. This allows us to show that the average $\tilde{d} = \frac{1}{t}\sum_{j=1}^{n} c_j$ falls close to its expectation d with high probability, giving Theorem 4.3.1.

Our re-collision probability bound is stated below:

Lemma 4.3.4 (Re-collision Probability Bound)**.** *Consider two agents a_1 and a_2 randomly walking on a two-dimensional torus of dimensions $\sqrt{A} \times \sqrt{A}$. Assume that a_1 and a_2 collide in round r. For any $m \geq 0$, let $\mathcal{W}$ be the m-step random walk performed by a_2 in rounds $r+1, ..., r+m$. Let $\mathcal{C}$ be the event that a_1 and a_2 collide again in round $r+m$. We have:*

$$\mathbb{P}[\mathcal{C}|\mathcal{W}] = O\left(\frac{1}{m+1} + \frac{1}{A}\right).$$

Lemma 4.3.4 Proof Outline.

Our proof of Lemma 4.3.4 in broken down into the following steps. See Figure 4-2 for a schematic of the proof.

1. In Lemma 4.3.5 we bound the probability that a single m-step random walk starting from some position ends at any other x or y position, conditioned on the number of steps that the walk takes in the x and y directions. The proof of this lemma breaks down into two cases:

[1] In fact, we prove a stronger result, giving a bound on the re-collision probability conditioned on the random walk taken by one of the agents in rounds $r+1, ..., r+m$. As discussed in Section 4.3.2, it will later be necessary to condition on this walk to ensure that the number of collisions between the agent and each other agent (i.e., $c_1, ..., c_n$) are independent.

- In Claim 4.3.6 we show that if the walk takes m_x steps in the x direction and m_y steps in the y direction and does not fully 'wrap around' the torus, the probability that it ends at any x position can be bounded by $O\left(\frac{1}{\sqrt{m_x}}\right)$. The probability that it ends at any y position can similarly be bounded by $O\left(\frac{1}{\sqrt{m_y}}\right)$.
- In Claim 4.3.7 we handle the case when the walk does wrap fully around the torus one or more times, showing that this possibility adds at most an additional $O\left(\frac{1}{\sqrt{A}}\right)$ factor to the probability of ending at any x or y position on a $\sqrt{A} \times \sqrt{A}$ torus. This gives an overall bound on the probability of ending at any x position of $O\left(\frac{1}{\sqrt{m_x}} + \frac{1}{\sqrt{A}}\right)$, and an analogous bound on the probability of ending at any y position of $O\left(\frac{1}{\sqrt{m_y}} + \frac{1}{\sqrt{A}}\right)$.

2. In Corollary 4.3.8 we show that, since movement in the x and y directions are independent, the probability of a single walk ending at any position after taking m_x steps in the x direction and m_y steps in the y direction is:

$$O\left(\left[\frac{1}{\sqrt{m_x}} + \frac{1}{\sqrt{A}}\right] \cdot \left[\frac{1}{\sqrt{m_y}} + \frac{1}{\sqrt{A}}\right]\right) = O\left(\frac{1}{\sqrt{m_x \cdot m_y}} + \frac{1}{A}\right).$$

3. In Lemma 4.3.9 we show that, with high probability, an m-step walk takes $\Theta(m)$ steps in both the x and y directions. Combined with Corollary 4.3.8, this yields an unconditional bound of $O\left(\frac{1}{m} + \frac{1}{A}\right)$ on the probability that a single random walk starting from some position ends at any other particular position after m steps.

4. Finally, we bound the probability that two walks re-collide after m steps, conditioned on the path of one of these walks, giving Lemma 4.3.4. Fixing this path fixes a position (x_{end}, y_{end}) that the agent is located in at round $r + m$. For a re-collision to occur, the other agent must also be located at this position at round $r + m$. We bound the probability of this event directly with the single walk bound of Lemma 4.3.9.

We begin with our bound on the probability of a single walk ending at any x and y position, conditioned on the number number of steps that it takes in each direction.

Lemma 4.3.5. *Consider an agent a_1 randomly walking on a two-dimensional torus of dimensions $\sqrt{A} \times \sqrt{A}$ which is at position (x_{begin}, y_{begin}) in round r. For any $m \geq 0$ and any position (x_{end}, y_{end}), let $\mathcal{C}_x$ be the event that a_1 has x position x_{end} at round*

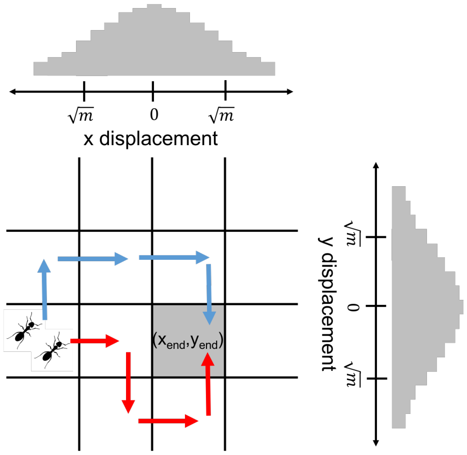

Figure 4-2: A schematic of the proof of Lemma 4.3.4.

$r+m$ and let $\mathcal{C}_y$ be the event that a_1 has y position y_{end} at round $r+m$. Let M_x, M_y be random variables giving the number of steps that a_1 takes in the x and y directions respectively in rounds $r+1, ...r+m$. For any $m_x, m_y \in \{0, ..., m\}$:

$$\mathbb{P}[\mathcal{C}_x | M_x = m_x] = O\left(\frac{1}{\sqrt{m_x+1}} + \frac{1}{\sqrt{A}}\right)$$

and

$$\mathbb{P}[\mathcal{C}_y | M_y = m_y] = O\left(\frac{1}{\sqrt{m_y+1}} + \frac{1}{\sqrt{A}}\right).$$

Proof. We focus on bounding $\mathbb{P}\left[\mathcal{C}_x | M_x = m_x\right]$. The bound on $\mathbb{P}\left[\mathcal{C}_y | M_y = m_y\right]$ follows from an identical proof. We split our analysis into two cases. Let $\delta_x = x_{end} - x_{begin}$ be the change in x position required for $\mathcal{C}_x$ to occur. Let $\mathcal{C}_x^1$ be the event that a_1 has total x displacement δ_x from round r to round $r+m$ (and so is at x position x_{end} in round $r+m$). Let $\mathcal{C}_x^2$ be the event that the agent is at x position x_{end} in round $r+m$ but *does not* have displacement δ_x. This requires that the agent 'wraps around' the torus, ending at x_{end} despite moving further than δ_x. We can write:

$$\mathbb{P}[\mathcal{C}_x | M_x = m_x] = \mathbb{P}[\mathcal{C}_x^1 | M_x = m_x] + \mathbb{P}[\mathcal{C}_x^2 | M_x = m_x]. \tag{4.1}$$

We bound the probabilities of $\mathcal{C}_x^1$ and $\mathcal{C}_x^2$ separately.

Claim 4.3.6 (Collision Probability Without Wraparound)**.**

$$\mathbb{P}[\mathcal{C}_x^1 | M_x = m_x] = O\left(\frac{1}{\sqrt{m_x+1}}\right).$$

Proof. We can write the x displacement of a_1 as $\sum_{j=1}^{m_x} s_j$ where s_j the direction of the agent's j^{th} step in the x direction. Each s_j is an independent random variable equal to 1 with probability 1/2 and -1 with probability 1/2. With this notation we can compute:

$$\mathbb{P}[\mathcal{C}_x^1 | M_x = m_x] = \mathbb{P}\left[\left(\sum_{j=1}^{m_x} s_j\right) = \delta_x | M_x = m_x\right] = \binom{m_x}{\frac{m_x+\delta_x}{2}} \left(\frac{1}{2}\right)^{m_x} \tag{4.2}$$

where we use the convention that the binomial coefficient equals 0 if $\frac{m_x+\delta_x}{2}$ is not a positive integer. For any δ_x we have $\binom{m_x}{\frac{m_x+\delta_x}{2}} \leq \binom{m_x}{\lfloor m_x/2 \rfloor} = \frac{m_x!}{\lfloor \frac{m_x}{2} \rfloor! \cdot \lceil \frac{m_x}{2} \rceil!}$ and by Stirling's approximation, for any $n > 0$, $n! = \sqrt{2\pi n}\left(\frac{n}{e}\right)^n \left(1 + O\left(\frac{1}{n}\right)\right)$, which gives:

$$\mathbb{P}[\mathcal{C}_x^1 | M_x = m_x] = O\left(\frac{1}{\sqrt{m_x+1}}\right),$$

and so the claim. Note that we use $m_x + 1$ instead of m_x in the denominator so that the bound is meaningful in the case when $m_x = 0$. □

We next show:

Claim 4.3.7 (Collision Probability with Wraparound)**.**

$$\mathbb{P}\left[\mathcal{C}_x^2 | M_x = m_x\right] = O\left(\frac{1}{\sqrt{m_x+1}} + \frac{1}{\sqrt{A}}\right)$$

Proof. In order for $\mathcal{C}_x^2$ to occur, a_1 must have x position x_{end} after round $r + m$ but not have total displacement δ_x. In particular, a_1's displacement must differ from δ_x by a nonzero integer multiple of $\sqrt{A}$ – the side length of the torus. We can thus write, letting $\mathbb{Z} \setminus 0$ denote the set of nonzero integers:

$$\begin{aligned}\mathbb{P}\left[\mathcal{C}_x^2 | M_x = m_x\right] &= \sum_{c \in \mathbb{Z} \setminus 0} \mathbb{P}\left[\left(\sum_{j=1}^{m_x} s_j\right) = \delta_x + c\sqrt{A} | M_x = m_x\right] \\ &= \left(\frac{1}{2}\right)^{m_x} \cdot \sum_{c \in \mathbb{Z} \setminus \{0, \pm 1\}} \binom{m_x}{\frac{m_x+\delta_x+c\sqrt{A}}{2}} + O\left(\frac{1}{\sqrt{m_x+1}}\right). \end{aligned} \tag{4.3}$$

To obtain (4.3) we bound the $c = \pm 1$ terms $\mathbb{P}\left[\left(\sum_{j=1}^{m_x} s_j\right) = \delta_x \pm \sqrt{A} | M_x = m_x\right]$ by $O\left(\frac{1}{\sqrt{m_x+1}}\right)$ using our bound on $\mathcal{C}_x^1$ given in Claim 4.3.6, which can easily be seen to hold for any x displacement, and in particular, for $\delta_x \pm \sqrt{A}$.

Roughly, we will upper bound the first term of (4.3) by the probability that the agent ends at *any* x position in round $r + m$. Since there are $\sqrt{A}$ such positions, this probability is thus bounded by $O\left(\frac{1}{\sqrt{A}}\right)$. Formally, consider any $i \in [1, ..., \sqrt{A} - 1]$ and let $\mathcal{D}_x^i$ be the event that the walk is i steps clockbywise from x_{end} after taking M_x steps. We can write:

$$\mathbb{P}[\mathcal{D}_x^i | M_x = m_x] = \left(\frac{1}{2}\right)^{m_x} \cdot \sum_{c \in \mathbb{Z}} \binom{m_x}{\frac{m_x+\delta_x+i+c\sqrt{A}}{2}}. \tag{4.4}$$

Now, since $|\delta_x| < \sqrt{A}$ and $i < \sqrt{A}$, for $c \geq 2$, $\delta_x + i + (c-1)\sqrt{A}$ is closer to 0 than $\delta_x + c\sqrt{A}$. So, as long as $\frac{m_x+\delta_x+i+c\sqrt{A}}{2}$ is an integer, $\binom{m_x}{\frac{m_x+\delta_x+i+(c-1)\sqrt{A}}{2}} \geq \binom{m_x}{\frac{m_x+\delta_x+c\sqrt{A}}{2}}$. Similarly, for $c \leq -2$, $\delta_x + i + c\sqrt{A}$ is closer to 0 than $\delta_x + c\sqrt{A}$. So as long as $\frac{m_x+\delta_x+i+c\sqrt{A}}{2}$ is an integer, $\binom{m_x}{\frac{m_x+\delta_x+i+c\sqrt{A}}{2}} \geq \binom{m_x}{\frac{m_x+\delta_x+c\sqrt{A}}{2}}$.

Let $\mathcal{E}_{i,c}$ equal 1 if $\frac{m_x+\delta_x+i+c\sqrt{A}}{2}$ is an integer and 0 otherwise. By the above bounds we have:

$$\begin{aligned}\sum_{c \in \mathbb{Z} \setminus \{0,\pm 1\}} \mathcal{E}_{i,c} \cdot \binom{m_x}{\frac{m_x+\delta_x+c\sqrt{A}}{2}} &\leq \sum_{c \in \mathbb{Z} \setminus \{0,-1\}} \mathcal{E}_{i,c} \cdot \binom{m_x}{\frac{m_x+\delta_x+i+c\sqrt{A}}{2}} \\ &\leq \sum_{c \in \mathbb{Z}} \mathcal{E}_{i,c} \cdot \binom{m_x}{\frac{m_x+\delta_x+i+c\sqrt{A}}{2}}.\end{aligned}$$

Combining with (4.4) and using the fact that the $\mathcal{D}_x^i$ events are disjoint, so the sum of their probabilities is at most 1, we have:

$$\begin{aligned}\sum_{i=1}^{\sqrt{A}-1} \mathbb{P}\left[\mathcal{D}_x^i | M_x = m_x\right] &\leq 1 \\ \sum_{i=1}^{\sqrt{A}-1} \left[\left(\frac{1}{2}\right)^{m_x} \cdot \sum_{c \in \mathbb{Z}} \mathcal{E}_{i,c} \cdot \binom{m_x}{\frac{m_x+\delta_x+i+c\sqrt{A}}{2}}\right] &\leq 1 \\ \sum_{i=1}^{\sqrt{A}-1} \left[\left(\frac{1}{2}\right)^{m_x} \sum_{c \in \mathbb{Z} \setminus \{0,\pm 1\}} \mathcal{E}_{i,c} \cdot \binom{m_x}{\frac{m_x+\delta_x+c\sqrt{A}}{2}}\right] &\leq 1 \\ \left(\frac{1}{2}\right)^{m_x} \sum_{c \in \mathbb{Z} \setminus \{0,\pm 1\}} \left[\left(\sum_{i=1}^{\sqrt{A}-1} \mathcal{E}_{i,c}\right) \cdot \binom{m_x}{\frac{m_x+\delta_x+c\sqrt{A}}{2}}\right] &\leq 1.\end{aligned}$$

Now, for all c, $\sum_{i=1}^{\sqrt{A}-1} \mathcal{E}_{i,c} = \Theta\left(\sqrt{A}\right)$ since $\frac{m_x+\delta_x+i+c\sqrt{A}}{2}$ is integral for half the possible $i \in [1, ..., \sqrt{A}-1]$. Rearranging, we thus have $\left(\frac{1}{2}\right)^{m_x} \sum_{c\in\mathbb{Z}\setminus\{0,\pm 1\}} \binom{m_x}{\frac{m_x+\delta_x+c\sqrt{A}}{2}} = O\left(\frac{1}{\sqrt{A}}\right)$. Plugging this back into (4.3):

$$\mathbb{P}\left[\mathcal{C}_x^2 | M_x = m_x\right] = O\left(\frac{1}{\sqrt{m_x+1}} + \frac{1}{\sqrt{A}}\right),$$

which gives the claim. □

Plugging the bounds of Claims 4.3.6 and 4.3.7 into (4.1) we have:

$$\begin{aligned}\mathbb{P}\left[\mathcal{C}_x | M_x = m_x\right] = \mathbb{P}\left[\mathcal{C}_x^1 | M_x = m_x\right] + \mathbb{P}\left[\mathcal{C}_x^2 | M_x = m_x\right]\\ O\left(\frac{1}{\sqrt{m_x+1}} + \frac{1}{\sqrt{A}}\right),\end{aligned}$$

which gives the lemma. □

Since an agent's movements in the x and y directions are independent, Lemma 4.3.5 yields:

Corollary 4.3.8. *Consider an agent a_1 randomly walking on a two-dimensional torus of dimensions $\sqrt{A} \times \sqrt{A}$ which is at position (x_{begin}, y_{begin}) in round r. For any $m \geq 0$ and any position (x_{end}, y_{end}), let $\mathcal{C}$ be the event that a_1 has x position (x_{end}, y_{end}) at round $r+m$. Let M_x, M_y be random variables giving the number of steps that a_1 takes in the x and y directions respectively in rounds $r+1, ...r+m$. For any $m_x, m_y \in \{0, ..., m\}$:*

$$\mathbb{P}[\mathcal{C} | M_x = m_x, M_y = m_y] = O\left(\frac{1}{\sqrt{(m_x+1)(m_y+1)}} + \frac{1}{A}\right).$$

Proof. Let $\mathcal{C}_x$ and $\mathcal{C}_y$ be as defined in Lemma 4.3.5. All steps are chosen independently, so conditioned on $M_x = m_x$ and $M_y = m_y$, $\mathcal{C}_x$ and $\mathcal{C}_y$ are independent. We can thus compute:

$$\begin{aligned}\mathbb{P}\left[\mathcal{C} | M_x = m_x, M_y = m_y\right] &= \mathbb{P}\left[\mathcal{C}_x \text{ and } \mathcal{C}_y | M_x = m_x, M_y = m_y\right]\\ &= \mathbb{P}\left[\mathcal{C}_x | M_x = m_x\right] \cdot \mathbb{P}\left[\mathcal{C}_y | M_y = m_y\right].\end{aligned}$$

Applying the bounds from Lemma 4.3.5 we have:

$$\begin{aligned}\mathbb{P}\left[\mathcal{C}|M_x = m_x, M_y = m_y\right] &= O\left(\left[\frac{1}{\sqrt{m_x+1}} + \frac{1}{\sqrt{A}}\right] \cdot \left[\frac{1}{\sqrt{m_y+1}} + \frac{1}{\sqrt{A}}\right]\right) \\ &= O\left(\frac{1}{\sqrt{(m_x+1)(m_y+1)}} + \frac{1}{A}\right),\end{aligned}$$

which gives the corollary. □

Using Corollary 4.3.8 we can give an unconditional bound on the collision probability by showing that, with high probability, $m_x = \Theta(m)$ and $m_y = \Theta(m)$.

Lemma 4.3.9 (Single Random Walk Collision Probability)**.** *Consider an agent a_1 randomly walking on a two-dimensional torus of dimensions $\sqrt{A} \times \sqrt{A}$ which is at position (x_{begin}, y_{begin}) in round r. For any $m \geq 0$ and any position (x_{end}, y_{end}), let $\mathcal{C}$ be the event that a_1 has x position (x_{end}, y_{end}) at round $r+m$.*

$$\mathbb{P}[\mathcal{C}] = O\left(\frac{1}{m+1} + \frac{1}{A}\right).$$

Proof. As in Corollary 4.3.8, let M_x, M_y be random variables giving the number of steps that a_1 takes in the x and y directions respectively in rounds $r+1, ...r+m$. Since direction is chosen independently and uniformly at random for each step, $\mathbb{E}[M_x] = \mathbb{E}[M_y] = m/2$. By a standard Chernoff bound:

$$\mathbb{P}[M_x \leq m/4] \leq 2e^{-(1/2)^2 \cdot m/4} = O\left(\frac{1}{m+1}\right).$$

(Again writing $m+1$ instead of m to cover the $m = 0$ case). An identical bound holds for M_y. Thus, by a union bound, except with probability $O\left(\frac{1}{m+1}\right)$ both M_x and M_y are $\geq m/4$. Applying Corollary 4.3.8 we have:

$$\begin{aligned}\mathbb{P}\left[\mathcal{C}\right] = \mathbb{P}\left[\mathcal{C}|M_x \geq m/4 \text{ and } M_y \geq m/4\right] \cdot \mathbb{P}\left[M_x \geq m/4 \text{ and } M_y \geq m/4\right] \\ + \mathbb{P}\left[\mathcal{C}|M_x < m/4 \text{ or } M_y < m/4\right] \cdot \mathbb{P}\left[M_x < m/4 \text{ or } M_y < m/4\right] \\ \leq \mathbb{P}\left[\mathcal{C}|M_x \geq m/4 \text{ and } M_y \geq m/4\right] + \mathbb{P}\left[M_x < m/4 \text{ or } M_y < m/4\right] \\ = O\left(\frac{1}{\sqrt{(m/4+1)(m/4+1)}} + \frac{1}{A} + \frac{1}{m+1}\right) \\ = O\left(\frac{1}{m+1} + \frac{1}{A}\right),\end{aligned}$$

which gives the lemma. □

We note that Lemma 4.3.9 immediately gives a bound of $O\left(\frac{1}{m+1}+\frac{1}{A}\right)$ on the probability that a single random walk returns to its origin (equalizes) after m steps. In fact a slightly stronger bound can be shown in this special case:

Corollary 4.3.10 (Equalization Probability Bound)**.** *Consider agent a_1 randomly walking on a two-dimensional torus of dimensions $\sqrt{A}\times\sqrt{A}$. If a_1 is located at position p after round r, for any even $m \geq 0$, the probability that a_1 is again at position p after round $r+m$ is $\Theta\left(\frac{1}{m+1}\right)+O\left(\frac{1}{A}\right)$. For odd m the probability is* 0.

Proof. The corollary has a $\Theta\left(\frac{1}{m+1}\right)$ bound instead of the $O\left(\frac{1}{m+1}\right)$ bound which would be given by directly applying Lemma 4.3.9. To obtain the stronger bound, simply note that when bounding the equalization probability, we have $\delta_x = 0$ (where δ_x is as defined in the proof of Lemma 4.3.5). As long as m_x is even, the bound in Claim 4.3.6 becomes $\Theta\left(\frac{1}{\sqrt{m_x+1}}\right)$. The remainder of the proof goes through unchanged, after noting that if m is even, M_x and M_y are both even with $\Theta(1)$ probability.

□

We finally use Lemma 4.3.9 to directly bound the probability of two random walks re-colliding after m steps, conditioned on the path of one of these walks. This yields our main re-collision probability bound, Lemma 4.3.4.

Proof of Lemma 4.3.4. Consider any m-step path w. Let (x_{end}, y_{end}) be the last position in w. Then, conditioned on $\mathcal{W} = w$, a_2 is at (x_{end}, y_{end}) in round $r+m$. Thus, conditioned on $\mathcal{W} = w$, a re-collision occurs in this round if and only if a_1 is also located at (x_{end}, y_{end}). Lemma 4.3.9 gives a bound on this probability, which yields the lemma. □

4.3.5 Collision Moment Bound

With Lemma 4.3.4 in hand, we can prove our collision moment bound, which we will use to show that the number of collisions an agent sees concentrates strongly around its expectation. Our moment bound is:

Lemma 4.3.11 (Collision Moment Bound)**.** *Let $\mathcal{W}$ be the t-step random walk that an agent executing Algorithm 1 takes. For $j \in [1, ..., n]$, let $\bar{c}_j \overset{\text{def}}{=} c_j - \mathbb{E}[c_j|\mathcal{W}]$ and assume $t \leq A$. There is some fixed constant w such that for any integer $k \geq 1$,*

$$\mathbb{E}\left[\bar{c}_j^k|\mathcal{W}\right] \leq \frac{tw^k}{A}\cdot k!\log^k(2t).$$

Note that, again, we condition the random walk taken by one of the agents. When $k = 2$, Lemma 4.3.11 gives a bound on the *variance* of c_j, which can be used to show that c_j falls close to its mean with good probability. By bounding the k^{th} moment $\mathbb{E}[\bar{c}_j^k|\mathcal{W}]$ for all k, we are able to show even stronger concentration results.

Our proof of Lemma 4.3.11 breaks down into the following steps:

1. In Lemma 4.3.12 we bound the probability that an agent collides with any other particular agent j at least once during the execution of Algorithm 1 (i.e., that $c_j \geq 1$), using a simple linearity of expectation argument.

2. In Claims 4.3.13 and 4.3.14 we bound the k^{th} moment of c_j for all $k \geq 1$ conditioned on $c_j = 0$ (i.e., there is no collision with agent a_j) and on $c_j \geq 1$ (i.e., there is at least one collision with agent a_j. The $c_j \geq 1$ case uses the re-collision probability bound of Lemma 4.3.4.

3. We combine the above results to give our final moment bound for each c_j, yielding Lemma 4.3.11.

Lemma 4.3.12 (First Collision Probability)**.** *Let $\mathcal{W}$ be the t-step random walk that an agent executing Algorithm 1 takes. For all $j \in [1, ..., n]$,*

$$\mathbb{P}\left[c_j \geq 1|\mathcal{W}\right] \leq \frac{t}{A}.$$

Proof. Using the fact that c_j is identically distributed for all j, and applying Lemma 4.3.2,

$$\mathbb{E}[\tilde{d}|\mathcal{W}] = d = \frac{1}{t} \cdot \mathbb{E}\left[\sum_{i=1}^{n} c_i \middle| \mathcal{W}\right] = \frac{n}{t} \cdot \mathbb{E}[c_j|\mathcal{W}] = \frac{n}{t} \cdot \mathbb{P}\left[c_j \geq 1|\mathcal{W}\right] \cdot \mathbb{E}[c_j|\mathcal{W}, c_j \geq 1]$$

$$\frac{n}{A} = \frac{n}{t} \cdot \mathbb{P}\left[c_j \geq 1|\mathcal{W}\right] \cdot \mathbb{E}[c_j|\mathcal{W}, c_j \geq 1].$$

Rearranging and noting that $\mathbb{E}[c_j|\mathcal{W}, c_j \geq 1] \geq 1$ gives:

$$\mathbb{P}\left[c_j \geq 1|\mathcal{W}\right] = \frac{t}{A \cdot \mathbb{E}[c_j|\mathcal{W}, c_j \geq 1]} \leq \frac{t}{A}. \tag{4.5}$$

□

We next give a simple bound on the moments of c_j conditioned on $c_j = 0$.

Claim 4.3.13. *Let $\mathcal{W}$ be the t-step random walk that an agent executing Algorithm 1 takes. For $j \in [1, ..., n]$, let $\bar{c}_j \overset{\text{def}}{=} c_j - \mathbb{E}[c_j|\mathcal{W}]$ and assume $t \leq A$. For any integer*

$k \geq 1$,

$$\mathbb{E}\left[\bar{c}_j^k | \mathcal{W}, c_j = 0\right] \leq \frac{t}{A}.$$

Proof. By the argument given in Lemma 4.3.2, $\mathbb{E}[c_j|\mathcal{W}] = \sum_{r=1}^t \mathbb{E}[c_j(t)|\mathcal{W}] = \frac{t}{A}$. We thus have: $\mathbb{E}\left[\bar{c}_j^k|\mathcal{W}, c_j = 0\right] = (0 - \mathbb{E}[c_j|\mathcal{W}])^k \leq (t/A)^k$. Further since $t \leq A$ by assumption, $t/A \leq 1$ and we can loosely bound $(t/A)^k \leq \frac{t}{A}$ for all $k \geq 1$, giving the claim. □

We next use the re-collision probability bound of Lemma 4.3.4 to bound the moments of c_j conditioned on $c_j \geq 1$.

Claim 4.3.14. *Let $\mathcal{W}$ be the t-step random walk that an agent executing Algorithm 1 takes. For $j \in [1, ..., n]$, let $\bar{c}_j \overset{\text{def}}{=} c_j - \mathbb{E}[c_j|\mathcal{W}]$ and assume $t \leq A$. There is some fixed constant w such that for any integer $k \geq 1$,*

$$\mathbb{E}\left[\bar{c}_j^k|\mathcal{W}, c_j \geq 1\right] \leq w^k \cdot k! \log^k(2t).$$

Proof. Since $\mathbb{E}[c_j|\mathcal{W}] = \frac{t}{A} \leq 1$, we have $\mathbb{E}\left[\bar{c}_j^k|\mathcal{W}, c_j \geq 1\right] \leq \mathbb{E}\left[c_j^k|\mathcal{W}, c_j \geq 1\right]$. So to prove the lemma, it just suffices to show that $\mathbb{E}\left[c_j^k|\mathcal{W}, c_j \geq 1\right] \leq k! w^k \log^k(2t)$ for some w and all $k \geq 1$.

Let t' be the first time in which there is a collision with agent j, and $t' = 1$ if there is no such collision. We split c_j over rounds as $c_j = \sum_{r=t'}^t c_j(r) \leq \sum_{r=t'}^{t'+t-1} c_j(r)$. Where we simply define $c_j(r) = 0$ for any $r > t$. To simplify notation we relabel round t' round 1 and so round $t' + t - 1$ becomes round t. After this relabeling, conditioned on $c_j \geq 1$, we have $c_j(1) = 1$. Expanding c_j^k out fully using the summation:

$$\begin{aligned}\mathbb{E}\left[c_j^k|\mathcal{W}, c_j \geq 1\right] &= \mathbb{E}\left[\sum_{r_1=1}^t \sum_{r_2=1}^t ... \sum_{r_k=1}^t c_j(r_1)c_j(r_2)...c_j(r_k)\middle|\mathcal{W}, c_j \geq 1\right] \\ &= \sum_{r_1=1}^t \sum_{r_2=1}^t ... \sum_{r_k=1}^t \mathbb{E}\left[c_j(r_1)c_j(r_2)...c_j(r_k)|\mathcal{W}, c_j \geq 1\right].\end{aligned}$$

$\mathbb{E}\left[c_j(r_1)c_j(r_2)...c_j(r_k)|\mathcal{W}, c_j \geq 1\right]$ is just the probability that the two agents collide in each of rounds $r_1, r_2, ..., r_k$, conditioned on the walk $\mathcal{W}$ and that $c_j(1) = 1$. Assume without loss of generality that $r_1 \leq r_2 \leq ... \leq r_k$. By Lemma 4.3.4 and the fact that $c_j(1) = 1$, for some fixed w we can bound this probability $\leq \frac{w^k}{r_1(r_2-r_1+1)(r_3-r_2+1)...(r_k-r_{k-1}+1)}$. Here we use the assumption that $t \leq A$ so the $O\left(\frac{1}{A}\right)$ term is absorbed into the $O\left(\frac{1}{m+1}\right)$ term in Lemma 4.3.4. We then rewrite, by linearity of expectation:

$$\mathbb{E}\left[c_j^k|\mathcal{W}, c_j \geq 1\right] \leq k! \sum_{r_1=1}^{t} \ldots \sum_{r_k=r_{k-1}}^{t} \frac{w^k}{r_1(r_2-r_1+1)\ldots(r_k-r_{k-1}+1)}.$$

The $k!$ comes from the fact that in this sum we have only ordered k-tuples and so need to multiple by $k!$ to account for the fact that the original sum is over unordered k-tuples. We can bound:

$$\sum_{r_k=r_{k-1}}^{t} \frac{1}{r_k-r_{k-1}+1} = 1+\frac{1}{2}+\ldots+\frac{1}{t} = O(\log 2t),$$

so rearranging the sum and simplifying gives:

$$\begin{aligned}\mathbb{E}\left[c_j^k|\mathcal{W}, c_j \geq 1\right] &\leq k!w^k \sum_{r_1=1}^{t} \frac{1}{r_1} \sum_{r_2=r_1}^{t} \frac{1}{r_2-r_1+1} \ldots \sum_{r_k=r_{k-1}}^{t} \frac{1}{r_k-r_{k-1}+1}\\ &\leq k!w^k \sum_{r_1=1}^{t} \ldots \sum_{r_{k-1}=r_{k-2}}^{t} \frac{1}{r_{k-2}-r_{k-1}+1} \cdot O(\log 2t).\end{aligned}$$

We repeat this argument for each level of summation replacing $\sum_{r_i=r_{i-1}}^{t} \frac{1}{r_i-r_{i-1}+1}$ with $O(\log 2t)$. Iterating through the k levels gives

$$\mathbb{E}\left[c_j^k|\mathcal{W}, c_j \geq 1\right] \leq k!w^k \log^k 2t,$$

after w is adjusted using the constant in the $O(\log 2t)$ term, establishing the claim. □

Finally, we combine claims 4.3.13 and 4.3.14 with the first collision probability bound of Lemma 4.3.12 to prove our main moment bound, Lemma 4.3.11.

Proof of Lemma 4.3.11. We expand:

$$\mathbb{E}[\bar{c}_j^k|\mathcal{W}] = \mathbb{P}[c_j \geq 1|\mathcal{W}] \cdot \mathbb{E}[\bar{c}_j^k|\mathcal{W}, c_j \geq 1] + \mathbb{P}[c_j = 0|\mathcal{W}] \cdot \mathbb{E}[\bar{c}_j^k|\mathcal{W}, c_j = 0].$$

By Lemma 4.3.12:

$$\mathbb{E}\left[\bar{c}_j^k|\mathcal{W}\right] \leq \frac{t}{A} \cdot \mathbb{E}\left[\bar{c}_j^k|\mathcal{W}, c_j \geq 1\right] + \mathbb{E}\left[\bar{c}_j^k|\mathcal{W}, c_j = 0\right].$$

Plugging in the bounds of Claims 4.3.13 and 4.3.14 we then have, for some fixed w

and all $k \geq 1$,

$$\begin{aligned}\mathbb{E}\left[\bar{c}_j^k|\mathcal{W}\right] &\leq \frac{tw^k}{A}k!\log^k(2t) + \frac{t}{A}\\ &\leq \frac{t(w+1)^k}{A}k!\log^k(2t),\end{aligned}$$

giving the lemma. □

As with Lemma 4.3.4, the techniques used in Lemma 4.3.11 can be applied to bounding the moments of the number of equalizations of a single random walk. We give two bounds that may be of independent interest. Note that the first bound is slightly tighter (by a $\log 2t$ factor) than what would be obtained simply by replacing the use of Lemma 4.3.4 with Corollary 4.3.10 in the proof of Claim 4.3.14.

Corollary 4.3.15 (Random Walk Visits Moment Bound)**.** *Consider an agent a_1 randomly walking on a two-dimensional $\sqrt{A} \times \sqrt{A}$ torus that is initially located at a uniformly random location and takes $t \leq A$ steps. Let c_j be the number of times that a_1 visits node j. There exists a fixed constant w such that for all $j \in [1, ...A]$ and all $k \geq 1$,*

$$\mathbb{E}\left[\bar{c}_j^k\right] \leq \frac{tw^k}{A} \cdot k!\log^{k-1}(2t).$$

Proof. We show that $\mathbb{P}[c_j \geq 1] = O\left(\frac{t}{A\log 2t}\right)$, strengthening Lemma 4.3.12 by a $\log 2t$ factor. Combining this stronger bound with Claims 4.3.13 and 4.3.14 and following the proof of Lemma 4.3.11 gives the result.

Let $c(r)$ be 1 if the agent visits node j in round r, and 0 otherwise. Due to the initial uniform distribution of the agent, by linearity of expectation:

$$\mathbb{E}[c] = \sum_{i=1}^{t}\mathbb{E}[c(r)] = \frac{t}{A}.$$

As in the proof of Lemma 4.3.12, we can rewrite this expectation as:

$$\mathbb{E}[c] = \frac{t}{A} = \mathbb{P}[c \geq 1] \cdot \mathbb{E}[c|c \geq 1].$$

To compute $\mathbb{E}[c|c \geq 1]$, we use Corollary 4.3.10 and linearity of expectation. Since $t \leq A$, the $O\left(\frac{1}{A}\right)$ term in Corollary 4.3.10 is absorbed into the $\Theta\left(\frac{1}{m+1}\right)$. Let $r \leq t$

be the first round that the agent visits node j. Then:

$$\mathbb{E}[c|c \geq 1] = \sum_{m=0}^{t-r} \Theta\left(\frac{1}{m+1}\right) = \Theta\left(\log(2(t-r+1))\right). \tag{4.6}$$

Further, the probability of the *first* visit to node j is in a given round can only decrease as the round number increases. So, at least $1/2$ of the time that $c \geq 1$, there is a visit in the first $t/2$ rounds (Note that we can assume $t \geq 2$ since if $t = 1$ we already have $\mathbb{E}[c|c \geq 1] = 1$). So, overall, by (4.6), $\mathbb{E}[c|c \geq 1] = \Theta\left(\log(2(t - t/2 + 1))\right) = \Theta\left(\log 2t\right)$. Using (4.5), $\mathbb{P}\left[c \geq 1\right] = O\left(\frac{t}{A \log 2t}\right)$, completing the proof. □

We have a similar bound on the number of returns to the agent's starting node.

Corollary 4.3.16 (Equalization Moment Bound). *Consider an agent a_1 randomly walking on a two-dimensional $\sqrt{A} \times \sqrt{A}$ torus. If a_1 takes $t \leq A$ steps and c is the number of times it returns to its starting position (the number of equalizations), there exists a fixed constant w such that for all $k \geq 1$,*

$$\mathbb{E}\left[\bar{c}^k\right] \leq k! w^k \log^k(2t).$$

Proof. This follows an identical proof to that of the moment bound given in Claim 4.3.14 for the number of collisions between two agents that are assumed to collide at least once: $\mathbb{E}\left[c_j^k | \mathcal{W}, c_j \geq 1\right] \leq k! w^k \log^k(2t)$. We simply replace the application of Lemma 4.3.4 with Corollary 4.3.10. □

4.3.6 Correctness of Encounter Rate With High Probability

Armed with Lemma 4.3.11 we can finally show that $\sum_{j=1}^n c_j$ concentrates strongly about its expectation. Since $\tilde{d} = \frac{1}{t}\sum_{j=1}^n c_j$, this is enough to prove the accuracy of encounter-rate-based density estimation (Algorithm 1). We first use Lemma 4.3.11 to give a standard 'sub-exponential' bound on the sum $\sum_{j=1}^n c_j$ (Corollary 4.3.17 below). It is in this step that we use that each c_j is independent conditioned on executing agent's random walk $\mathcal{W}$. We correspondingly remove this conditioning, giving an unconditioned bound.

Corollary 4.3.17 (Sub-exponential condition). *Assuming $t \leq A$, for some $b = \Theta(\log 2t)$, $\sigma^2 = \Theta(td\log^2 2t)$, and any λ with $|\lambda| \leq 1/b$ we have:*

$$\mathbb{E}\left[e^{\lambda\left(\sum_{j=1}^n c_j - \mathbb{E}\left[\sum_{j=1}^n c_j\right]\right)}\right] \leq e^{\frac{\lambda^2\sigma^2}{2-2b|\lambda|}}.$$

Proof. Let $\mathcal{W}$ be the t-step random walk that an agent executing Algorithm 1 takes. By Lemma 4.3.11, there exists some constant w such that for $\sigma^2 = \frac{w^2 t \log^2 2t}{A}$ and $b = w \log 2t$, $\bar{c}_j \stackrel{\text{def}}{=} c_j - \mathbb{E}[c_j|\mathcal{W}]$ satisfies:

$$\mathbb{E}\left[\bar{c}_j^k|\mathcal{W}\right] \leq \frac{1}{2}k!\sigma^2 b^{k-2}.$$

By Proposition 2.3 of [Wai15], this gives the sub-exponential moment bound: for any λ with $|\lambda| \leq 1/b$,

$$\mathbb{E}[e^{\lambda \bar{c}_j}|\mathcal{W}] \leq e^{\frac{\lambda^2\sigma^2}{2-2b|\lambda|}}.$$

Since each c_j is independent conditioned on the walk $\mathcal{W}$ this gives:

$$\mathbb{E}\left[e^{\lambda\left(\sum_{j=1}^n c_j - \mathbb{E}\left[\sum_{j=1}^n c_j|\mathcal{W}\right]\right)}|\mathcal{W}\right] = \mathbb{E}\left[\prod_{j=1}^n e^{\lambda\bar{c}_j}\middle|\mathcal{W}\right] = \prod_{j=1}^n \mathbb{E}\left[e^{\lambda\bar{c}_j}|\mathcal{W}\right] \leq e^{\frac{n\lambda^2\sigma^2}{2-2b|\lambda|}}.$$

The lemma follows by replacing σ^2 with $n\sigma^2 = \Theta(td\log^2 2t)$. Further, the above bound holds for all $\mathcal{W}$ and $\mathbb{E}[c_j|\mathcal{W}] = \mathbb{E}[c_j]$ (see Corollary 4.3.3). So, by the law of iterated expectation, we remove the conditioning on $\mathcal{W}$:

$$\mathbb{E}\left[e^{\lambda\left(\sum_{j=1}^n c_j - \mathbb{E}\left[\sum_{j=1}^n c_j\right]\right)}\right] = \mathbb{E}\left[\mathbb{E}\left[e^{\lambda\left(\sum_{j=1}^n c_j - \mathbb{E}\left[\sum_{j=1}^n c_j|\mathcal{W}\right]\right)}\middle|\mathcal{W}\right]\right] \leq e^{\frac{n\lambda^2\sigma^2}{2-2b|\lambda|}}.$$

□

We will employ a concentration bound for random variables satisfying such a sub-exponential condition:

Lemma 4.3.18 (Proposition 2.3 of [Wai15])**.** *Suppose that X satisfies $\mathbb{E}\left[e^{\lambda(X-\mathbb{E}[X])}\right] \leq e^{\frac{\lambda^2\sigma^2}{2-2b|\lambda|}}$ for any λ with $|\lambda| \leq 1/b$. Then for any $\Delta \geq 0$,*

$$\mathbb{P}\left[|X - \mathbb{E}[X]| \geq \Delta\right] \leq 2e^{-\frac{\Delta^2}{2(\sigma^2+b\Delta)}}.$$

Proof. This bound is given in Proposition 2.3 of [Wai15]. We include a full proof for completeness. We have $\mathbb{P}\left[|X - \mathbb{E}[X]| \geq \Delta\right] = \mathbb{P}\left[(X - \mathbb{E}[X]) \geq \Delta\right] + \mathbb{P}\left[(X - \mathbb{E}[X]) \leq -\Delta\right]$. We can bound these terms similarly. Focusing on the first, for any positive λ,

$$\mathbb{P}\left[(X - \mathbb{E}[X]) \geq \Delta\right] = \mathbb{P}\left[e^{\lambda(X-\mathbb{E}[X])} \geq e^{\lambda\Delta}\right].$$

By Markov's inequality and our moment bound, for any λ with $|\lambda| \leq 1/b$:

$$\mathbb{P}\left[e^{\lambda(X-\mathbb{E}[X])} \geq e^{\lambda\Delta}\right] \leq \mathbb{E}\left[e^{\lambda|X-\mathbb{E}[X]|}\right] \cdot e^{-\lambda\Delta} \leq e^{\left(\frac{\lambda^2\sigma^2}{2-2b|\lambda|}-\lambda\Delta\right)}.$$

We can set $\lambda = \frac{\Delta}{\sigma^2+b\Delta}$ and calculate:

$$\frac{\lambda^2\sigma^2}{2-2b|\lambda|} - \lambda\Delta = \frac{\Delta^2\sigma^2}{(\sigma^2+b\Delta)^2 \cdot \frac{2\sigma^2}{\sigma^2+b\Delta}} - \frac{\Delta^2}{\sigma^2+b\Delta} = -\frac{\Delta^2}{2(\sigma^2+b\Delta)}.$$

Which gives $\mathbb{P}\left[(X-\mathbb{E}[X]) \geq \Delta\right] \leq e^{-\frac{\Delta^2}{2(\sigma^2+b\Delta)}}$. We can bound $\mathbb{P}\left[(X-\mathbb{E}[X]) \leq -\Delta\right]$ in the same way by setting $\lambda = -\frac{\Delta}{\Delta b+\sigma^2}$, giving the Lemma. □

We conclude by proving our main theorem on the accuracy of random-walk-based density estimation:

Proof of Theorem 4.3.1. In Algorithm 1, $\tilde{d}$ is set to $\frac{1}{t}\sum_{j=1}^n c_j$. So the probability that $\tilde{d}$ falls within an ϵ multiplicative factor of its mean is the same as the probability that $\sum_{j=1}^n c_j$ falls within an ϵ multiplicative factor of its mean, which is equal to $t\,\mathbb{E}[\tilde{d}] = td$ by Corollary 4.3.3. By Corollary 4.3.17 and Lemma 4.3.18:

$$\begin{aligned}\delta &\overset{\text{def}}{=} \mathbb{P}\left[\left|\sum_{j=1}^n c_j - \mathbb{E}\left[\sum_{j=1}^n c_j\right]\right| \geq \epsilon\,\mathbb{E}\left[\sum_{j=1}^n c_j\right]\right] \\ &= \mathbb{P}\left[\left|\sum_{j=1}^n c_j - td\right| \geq \epsilon td\right] \leq 2e^{\Theta\left(-\frac{\epsilon^2t^2d^2}{2(td\log^2 2t+\epsilon td\log 2t)}\right)} = 2e^{\Theta\left(-\frac{\epsilon^2 td}{\log^2 2t}\right)},\end{aligned}$$

where the last equality follows since we restricting $\epsilon \leq 1$. The above thus gives $\frac{\epsilon^2 td}{\log^2 2t} = O\left(\log(1/\delta)\right)$ and so $\epsilon \leq c_1 \cdot \sqrt{\frac{\log(1/\delta)}{td}} \cdot \log 2t$ for some fixed constant c_1. This gives the first claim of the theorem: for any $\delta > 0$, with probability $1-\delta$,

$$\tilde{d} \in [(1-\epsilon)d, (1+\epsilon)d] \text{ for } \epsilon \leq c_1 \cdot \sqrt{\frac{\log(1/\delta)}{td}} \cdot \log 2t. \tag{4.7}$$

Given any fixed $\epsilon, \delta \in (0,1)$ we can also rearrange by solving (4.7) for t which gives: $\frac{t}{\log^2 2t} \geq \frac{c_1^2\log(1/\delta)}{\epsilon^2 d}$. If we set, for some constant c_3,

$$t = \frac{c_3 \cdot c_1^2 \log(1/\delta)[\log\log(1/\delta) + \log(1/d\epsilon)]^2}{d\epsilon^2},$$

we have $\log 2t = \log(2c_3) + c_4[\log\log(1/\delta) + \log(1/d\epsilon)]$ for some fixed constant c_4 which is independent of c_3. Thus, setting c_3 large enough gives $\frac{t}{\log^2 2t} \geq \frac{c_1^2\log(1/\delta)}{\epsilon^2 d}$.

Setting $c_2 = c_3 \cdot c_1^2$, this yields the second claim of the theorem:

$$\text{for any } \epsilon, \delta \in (0,1), \text{ if } A \geq t \geq \frac{c_2 \log(1/\delta)[\log\log(1/\delta) + \log(1/d\epsilon)]^2}{d\epsilon^2},$$

then $\tilde{d} \in [(1-\epsilon)d, (1+\epsilon)d]$ with probability $\geq 1-\delta$. □

4.4 Extensions to Other Regular Topologies

We now discuss extensions of our results to a broader set of graph topologies, demonstrating the generality of our local mixing analysis. We illustrate divergence between local and global mixing properties, which can have significant effects on random-walk-based algorithms

4.4.1 From Re-collision Bounds to Density Estimation

Our proofs for the two-dimensional torus are largely independent of graph structure, using just a re-collision probability bound (Lemma 4.3.4) and the regularity (uniform node degrees) of the grid, so agents remain uniformly distributed on the nodes in each round (see for example, Lemma 4.3.2). Hence, extending our results to other regular graphs primarily involves obtaining re-collision probability bounds for these graphs.

We consider agents on a graph with A nodes that execute analogously to Algorithm 1, stepping to a random neighbor in each round. Again, we focus on the multi-agent case but similar bounds (resembling Corollaries 4.3.15 and 4.3.16) hold for a single random walk. We start with a lemma that gives density estimation accuracy in terms of re-collision probability. This is a direct generalization of our grid analysis.

Lemma 4.4.1 (Re-collision Probability to Density Estimation Accuracy). *Consider a regular graph (uniform nodes degrees) with A nodes along with two agents a_1 and a_2 randomly walking on this graph. Assume that a_1 and a_2 collide in round r. Suppose that there exists some function $\beta(m)$ such that, for any $0 \leq m \leq t$, letting $\mathcal{W}$ be the random m-step path taken by a_2 in rounds $r+1, ..., r+m$, and $\mathcal{C}$ be the event that a_1 and a_2 re-collide in round $r+m$:*

$$Pr[\mathcal{C}|\mathcal{W}] = O(\beta(m)).$$

Let $B(t) \overset{\text{def}}{=} \sum_{m=0}^{t} \beta(m)$. After running for $t \leq A$ steps, Algorithm 1 returns $\tilde{d}$ such *that, for any $\delta > 0$, with probability $\geq 1 - \delta$,*

$$\tilde{d} \in [(1-\epsilon)d, (1+\epsilon)d] \text{ for } \epsilon = O\left(\sqrt{\frac{\log(1/\delta)}{td}} \cdot B(t)\right).$$

Note that in the special case of the two-dimensional torus, by Lemma 4.3.4, we can set $\beta(m) = 1/(m+1)$ and hence $B(t) = O(\log 2t)$, recovering Theorem 4.3.1.

Proof. Let $\mathcal{W}$ denote the t-step random walk of an agent executing Algorithm 1. $\mathbb{E}[\tilde{d}|\mathcal{W}] = d$ (Lemma 4.3.2) still holds as the regularity of the graph ensures that agents remain uniformly distributed on the nodes in every round (the stable distribution of any regular graph is the uniform distribution).

Further, following the moment calculations in Claim 4.3.14, $\mathbb{E}[c_j^k|\mathcal{W}, c_j \geq 1] \leq k!w^k B(t)^k$ for some constant w. Claim 4.3.13 and Lemma 4.3.12 still hold, giving that the bound of Lemma 4.3.11 holds unchanged:

$$\mathbb{E}[\bar{c}_j^k|\mathcal{W}] \leq \frac{tw^k}{A} \cdot k!B(t)^k.$$

As in Corollary 4.3.17, this gives that $\sum_{j=1}^n c_j$ satisfies the sub-exponential condition

$$\mathbb{E}\left[e^{\lambda\left(\sum_{j=1}^n c_j - \mathbb{E}\left[\sum_{j=1}^n c_j\right]\right)}\right] \leq e^{\frac{\lambda^2\sigma^2}{2-2b|\lambda|}}$$

for $b = \Theta(B(t))$ and $\sigma^2 = \Theta\left(tdB(t)^2\right)$. Plugging into Lemma 4.3.18 gives $\frac{\epsilon^2 td}{B(t)^2} = O(\log(1/\delta))$. Rearranging yields the result. □

4.4.2 Density Estimation on the Ring

We first consider the ring, where the following re-collision probability bound holds:

Lemma 4.4.2 (Re-collision Probability Bound – Ring)**.** *Consider two agents a_1 and a_2 randomly walking on a one-dimensional torus (a ring) with A nodes. Assume that a_1 and a_2 collide in round r. For any $m \geq 0$, let $\mathcal{W}$ be the m-step random walk performed by a_2 in rounds $r+1, ..., r+m$. Let $\mathcal{C}$ be the event that a_1 and a_2 collide again in round $r+m$. We have:*

$$\mathbb{P}[\mathcal{C}|\mathcal{W}] = O\left(\frac{1}{\sqrt{m+1}} + \frac{1}{A}\right).$$

Proof. This bound holds via an essentially identical proof to the bound on $\mathcal{C}_x$ given in Claim 4.3.5. An m-step random walk on a line ends at any position with probability

$O(1/\sqrt{m+1})$. On a ring with A nodes the slightly weaker bound of $O\left(\frac{1}{\sqrt{m+1}}+\frac{1}{A}\right)$ holds. □

Density Estimation Bound: For $m \leq A$, the $O\left(\frac{1}{A}\right)$ term in Lemma 4.4.2 is absorbed into the $O\left(\frac{1}{\sqrt{m+1}}\right)$ and one can show that $\sum_{m=0}^{t} 1/\sqrt{m+1} = \Theta(\sqrt{t})$. Plugging into Lemma 4.4.1, we obtain

$$\epsilon = O\left(\sqrt{\frac{\log(1/\delta)}{td}} \cdot \sqrt{t}\right) = O\left(\sqrt{\frac{\log(1/\delta)}{d}}\right).$$

Note that this bound does not depend on t. That is, since local mixing on the ring is much worse than on the torus, our general technique is not strong enough to show the convergence of random-walk-based density estimation.

We do note that we can give a density estimation accuracy bound by using an alternative analysis which bounds the variance of the collision count $c = \sum c_j$ and applies Chebyshev's inequality. This technique is very similar to what is used in our network size estimation bounds in Section 4.6.1. A similar analysis can be applied to the two-dimensional torus and the other graphs we consider, giving worse dependence on the failure probability δ, but slightly improved dependence on other parameters.

Theorem 4.4.3 (Alternative Accuracy Bound – Ring). *After running for t rounds, assuming $t \leq A^2$, an agent executing Algorithm 1 on a ring with A nodes returns $\tilde{d}$ such that, for any $\delta > 0$, with probability $\geq 1-\delta$,*

$$\tilde{d} \in [(1-\epsilon)d, (1+\epsilon)d] \textit{ for } \epsilon = O\left(\sqrt{\frac{1}{t^{1/2}d\delta}}\right).$$

This implies that, for any $\epsilon, \delta \in (0,1)$ if $t = \Omega\left(\frac{1}{(d\epsilon^2\delta)^2}\right)$, $\tilde{d} \in [(1-\epsilon)d, (1+\epsilon)d]$ with probability $\geq 1-\delta$.

Note that in Theorem 4.4.3, t is required to be quadratic in $\frac{1}{d\epsilon^2}$, rather than linear as in Theorem 4.3.1. The weakness of this bound is again due to the poor local mixing on the ring which means that, over t steps, we expect to see $\Theta(\sqrt{t})$ rather than $\Theta(\log t)$ repeat collisions with every agent interacted with.

Proof. We first note that, if we do not condition on the path of either agent, the collision count c_j with any agent j is identically distributed to the number of times that a single random walk of length $2t$, initially placed uniformly at random, visits some location on the ring. We gave a bound on the moments of this visit count on the two-dimensional torus in Corollary 4.3.15. A nearly identical analysis gives a similar bound on the ring. On the ring, the calculation of $\mathbb{E}[c|c \geq 1]$ in (4.6) becomes

$$\mathbb{E}[c_j|c_j \geq 1] = \sum_{m=0}^{t-r} \Theta\left(\frac{1}{\sqrt{m+1}}\right) = \Theta\left(\sqrt{t-r}\right)$$

where r is the round in which the first visit occurs. We again argue that $r < t/2$ with probability $1/2$, which lets us show that $\mathbb{P}[c_j \geq 1] = O\left(\frac{\sqrt{t}}{A}\right)$ and overall:

$$\mathbb{E}[\bar{c}_j^k] \leq \frac{tw^k}{A} k! \cdot t^{(k-1)/2}.$$

where $\bar{c}_j = c_j - \mathbb{E}[c_j]$. In particular, this gives the variance bound:

$$\mathbb{E}[\bar{c}_j^2] = O\left(\frac{t^{3/2}}{A}\right).$$

We then have, recalling that $c = \sum_{i=1}^n c_j$ and letting $\bar{c} = c - \mathbb{E}[c]$,

$$\begin{aligned}\mathbb{E}\left[\bar{c}^2\right] = \mathbb{E}\left[\left(\sum_{i=1}^n \bar{c}_j\right)^2\right] &= \sum_{i=1}^k \mathbb{E}\left[\bar{c}_j^2\right] + \sum_{i=1}^n \sum_{j=i+1}^n 2\,\mathbb{E}\left[\bar{c}_i \cdot \bar{c}_j\right] \\ &= O\left(\frac{nt^{3/2}}{A}\right) + \sum_{i=1}^n \sum_{j=i+1}^n 2\,\mathbb{E}[\bar{c}_i \cdot \bar{c}_j]. \qquad (4.8)\end{aligned}$$

Let $\mathcal{W}$ be the path taken by the central agent a and let $\mathcal{S}$ be the set of all possible instantiations of this path. $\bar{c}_i$ and $\bar{c}_j$ are independent conditioned on this path. So:

$$\begin{aligned}\mathbb{E}[\bar{c}_i \cdot \bar{c}_j] &= \sum_{w \in \mathcal{S}} \mathbb{P}[\mathcal{W} = w] \cdot \mathbb{E}[\bar{c}_i|\mathcal{W} = w] \cdot \mathbb{E}[\bar{c}_j|\mathcal{W} = w] \\ &= 0\end{aligned}$$

where we use that $\mathbb{E}[c_i] = td = \mathbb{E}[c_i|\mathcal{W} = w]$ for all w, and thus $\mathbb{E}[\bar{c}_i|\mathcal{W} = w] = 0$.

Plugging back into (4.8) gives $\mathbb{E}\left[\bar{c}^2\right] = O\left(\frac{nt^{3/2}}{A}\right) = O\left(dt^{3/2}\right)$. Recall that $\mathbb{E}[c] = dt$ and we thus have, via Chebyshev's inequality,

$$\delta \stackrel{\text{def}}{=} \mathbb{P}\left[|c - \mathbb{E}[c]| \geq \epsilon\,\mathbb{E}[c]\right] = O\left(\frac{dt^{3/2}}{(\epsilon dt)^2}\right) = O\left(\frac{1}{\epsilon^2 d\sqrt{t}}\right).$$

Rearranging gives $\epsilon = O\left(\sqrt{\frac{1}{t^{1/2}d\delta}}\right)$ and so the theorem. □

4.4.3 Density Estimation on k-Dimensional Tori

We next consider density estimation on k-dimensional tori for general $k \geq 3$. As k increases, local mixing becomes stronger, fewer re-collisions occur, and density estimation becomes easier. In fact, for any constant $k \geq 3$, although the torus still mixes slowly (with global mixing time on the order of $A^{2/k}$ [AF02]), density estimation is as accurate as on the complete graph! Throughout this section we assume that k is a small constant and so hide multiplicative factors in $f(k)$ for any function f in our asymptotic notation. We subscript the notation with k to make this clear.

Lemma 4.4.4 (Re-collision Probability Bound – High-Dimensional Torus)**.** *Consider two agents a_1 and a_2 randomly walking on a k-dimensional torus with A nodes. Assume that a_1 and a_2 collide in round r. For any $m \geq 0$, let $\mathcal{W}$ be the m-step random walk performed by a_2 in rounds $r+1, ..., r+m$. Let $\mathcal{C}$ be the event that a_1 and a_2 collide again in round $r+m$. We have:*

$$\mathbb{P}[\mathcal{C}|\mathcal{W}] = O_k\left(\frac{1}{(m+1)^{k/2}} + \frac{1}{A}\right).$$

Proof. We closely follow the proof of Lemma 4.3.4. In total, a_1 takes m steps: M_i in each dimension for $i \in [1, ..., k]$. Let $\mathcal{C}_i$ be the event that the agents have the same position in the i^{th} dimension in round $r+m$. By the analysis of Lemma 4.3.5,

$$\mathbb{P}[\mathcal{C}_i|M_i = m_i] = O\left(\frac{1}{\sqrt{m_i+1}} + \frac{1}{A^{1/k}}\right).$$

So, following the analysis of Corollary 4.3.8, since movement in each of the k directions is independent,

$$\mathbb{P}[\mathcal{C}|M_1 = m_1, ..., M_k = m_k] = O\left(\frac{1}{\sqrt{m_1+1}} + \frac{1}{A^{1/k}}\right) \cdot .... \cdot O\left(\frac{1}{\sqrt{m_k+1}} + \frac{1}{A^{1/k}}\right). \tag{4.9}$$

We can then remove the conditioning on $M_1, ..., M_k$ similarly to Lemma 4.3.9. In expectation, $M_i = m/k$. So by a Chernoff bound,

$$\mathbb{P}\left[M_i \leq \frac{m}{2k}\right] \leq 2e^{-(1/2)^2 \cdot m/3k} = O\left(\frac{1}{(m+1)^{k/2}}\right)$$

again assuming k is a small constant. Union bounding over all k dimensions, we have

$M_i \geq m/(2k)$ for all i except with probability $O\left(\frac{1}{(m+1)^{k/2}}\right)$ and hence by (4.9):

$$\mathbb{P}[\mathcal{C}] = O\left(\frac{1}{(m+1)^{k/2}}\right) + \left[O\left(\frac{1}{\sqrt{m/(2k)+1}} + \frac{1}{A^{1/k}}\right)\right]^k = O_k\left(\frac{1}{(m+1)^{k/2}} + \frac{1}{A}\right),$$

giving the lemma (again, asymptotic notation hides multiplicative factors in k since it is a constant). □

Density Estimation Bound: We can plug the bound of Lemma 4.4.4 into Lemma 4.4.1. For $t \leq A$ and $k \geq 3$,

$$\sum_{m=0}^{t}\left(\frac{1}{(m+1)^{k/2}} + \frac{1}{A}\right) < 1 + \sum_{m=0}^{\infty}\frac{1}{(m+1)^{k/2}} = O(1).$$

So we can set $B(t) = O_k(1)$ and have $\epsilon = O_k\left(\frac{\sqrt{\log(1/\delta)}}{td}\right)$. Rearranging, we require $t = O_k\left(\frac{\log(1/\delta)}{\epsilon^2 d}\right)$. This matches independent sampling up to constants and multiplicative factors in k.

4.4.4 Density Estimation on Regular Expanders

When a graph *does* mix well globally, it mixes well locally. An obvious example is the complete graph, on which random-walk-based and independent-sampling-based density estimation are equivalent. We extend this intuition to any regular expander. An expander is a graph whose random walk matrix has its second eigenvalue bounded away from 1, and so on which random walks mix quickly. Expanders are 'well-connected' graphs with many applications, including in the design of robust communication networks [BP73] and efficient sampling schemes [Gil98].

Lemma 4.4.5 (Re-collision Probability Bound – Regular Expander)**.** *Let G be a k-regular expander with A nodes and adjacency matrix $\mathbf{M}$. Let $\mathbf{W} = \frac{1}{k}\cdot\mathbf{M}$ be its random walk matrix, with eigenvalues $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_A$. Let $\lambda = \max\{|\lambda_2|, |\lambda_A|\} < 1$. Consider two agents a_1 and a_2 randomly walking on G. Assume that a_1 and a_2 collide in round r. For any $m \geq 0$, let $\mathcal{W}$ be the m-step random walk performed by a_2 in rounds $r+1, \ldots, r+m$. Let $\mathcal{C}$ be the event that a_1 and a_2 collide again in round $r+m$. We have:*

$$\mathbb{P}[\mathcal{C}|\mathcal{W}] \leq \lambda^m + 1/A.$$

Proof. Suppose that a_1 and a_2 collide at node i in round r. For any m-step path w on G, conditioning on $\mathcal{W} = w$ fixes a_2's position in round $r+m$, which we denote as node j. A re-collision occurs at around $r+m$ if and only if a_1 is also located at node j in this round. The probability of this event can be bounded by the maximum probability of a_1 being at any node j in round $r+m$ after starting from node i. This equals $\max_{j\in\{1,\ldots,|A|\}}(\mathbf{W}^m\mathbf{e}_i)_j$, which we can bound using the following lemma on how rapidly an expander random walk converges to its stable distribution:

Lemma 4.4.6 (See [Lov93])**.** *Let G be a k-regular expander with A nodes, adjacency matrix $\mathbf{M}$, and random walk matrix $\mathbf{W} = \frac{1}{k}\cdot\mathbf{M}$. Let $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_A$ be the eigenvalues of $\mathbf{W}$ and $\lambda = \max\{|\lambda_2|, |\lambda_A|\} < 1$. For each $1 \leq i, j \leq A$,*

$$\left|(\mathbf{W}^m \cdot \mathbf{e}_i)_j - \frac{1}{A}\right| \leq \lambda^m.$$

This gives $\max_{j\in\{1,\ldots,|A|\}}(\mathbf{W}^m\mathbf{e}_i)_j \leq \lambda^m + \frac{1}{A}$, giving the Lemma. □

Density Estimation Bound: Again, we plug Lemma 4.4.5 in Lemma 4.4.1, setting $B(t) = \sum_{m=0}^{t}\beta(m) \leq \frac{1}{1-\lambda} + t/A$. Assuming $t = O(A)$,

$$\epsilon = O\left(\sqrt{\frac{\log(1/\delta)}{td}}\cdot\left(\frac{1}{1-\lambda}+\frac{t}{A}\right)^2\right) = O\left(\sqrt{\frac{\log(1/\delta)}{td(1-\lambda)^2}}\right).$$

Rearranging, $t = O\left(\frac{\log(1/\delta)}{\epsilon^2 d(1-\lambda)^2}\right)$, matching independent sampling up to a factor of $O(1/(1-\lambda)^2)$.

4.4.5 Density Estimation k-Dimensional Hypercubes

Finally, we give bounds for a k-dimensional hypercube. Such a graph has $A = 2^k$ vertices mapped to the elements of $\{\pm 1\}^k$, with an edge between any two vertices that differ by hamming distance 1. The hypercube is relatively fast mixing. Its adjacency matrix eigenvalues are $[-k, -k+2, ..., k-2, k]$. Since it is bipartite, we can ignore the negative eigenvalues: to return to its origin, a random walk must take an even number of steps, so we need only need to consider the squared walk matrix $\mathbf{W}^2$, which has all positive eigenvalues. Applying Lemma 4.4.5 with $\lambda = \Theta(1-2/k) = \Theta(1-1/\log A)$, gives $t = O\left(\frac{\log(1/\delta)\log^2(A)}{\epsilon^2 d}\right)$. However, it is possible to remove the dependence on A via a more refined analysis – while the global mixing time of the graph increases as A grows, local mixing becomes stronger!

Lemma 4.4.7 (Re-collision Probability Bound – k-Dimensional Hypercube)**.** *Consider two agents a_1 and a_2 randomly walking on a k-dimensional hypercube with $A = 2^k$ vertices. Assume that a_1 and a_2 collide in round r. For any $m \geq 0$, let $\mathcal{W}$ be the m-step random walk performed by a_2 in rounds $r+1, ..., r+m$. Let $\mathcal{C}$ be the event that a_1 and a_2 collide again in round $r+m$. We have:*

$$\mathbb{P}[\mathcal{C}|\mathcal{W}] \leq (9/10)^{m-1} + \frac{1}{\sqrt{A}}.$$

Proof. A node of the hypercube can be represented as a k-bit string and each random walk step seen as choosing one of the bits uniformly at random and flipping it. Without loss of generality, assume that a_1 and a_2 collide at the node corresponding to bit string $[0, 0, ..., 0]$ in round r. For any m-step path w, conditioning on $\mathcal{W} = w$ fixes a_2's position in round $r+m$, which we denote as node p_{end}. We can write this position using its associate bit string as $p_{end} = [p_1, ..., p_k]$. A re-collision occurs at around $r+m$ if and only if a_1 is also located at p_{end} in this round. The probability of this event can be bounded by the maximum probability of a_1 being at any position $p_{end} = [p_1, ..., p_k]$ in round $r+m$ after starting from position $[0, 0, ..., 0]$. We bound this probability below, giving the lemma. □

Lemma 4.4.8. *Consider an agent a_1 randomly walking on a k-dimensional hypercube with $A = 2^k$ vertices, each labeled by a bit string in $\{0,1\}^k$. Assume that a_1 starts at position $[0, 0, ..., 0]$. For any $m \geq 0$ and position $p_{end} = [p_1, ..., p_k]$, let $\mathcal{C}$ be the event that a_1 is at position p_{end} after m steps. We have:*

$$\mathbb{P}[\mathcal{C}] \leq (9/10)^{m-1} + \frac{1}{\sqrt{A}}.$$

Proof. We will upper bound the probability that a_1 ends at any position p_{end}, denoted by the bit string $p_{end} = [p_1, ..., p_k]$ after m steps, assuming that m is even. If m is odd, the probability of ending at p_{end} after m steps is equal to the sum over all p'_{end} with hamming distance 1 from p_{end} of the probability of ending at p'_{end} after $m-1$ steps. There are k such positions. Further, the probability of moving from any p'_{end} to p_{end} in a single random-walk step is $1/k$. Thus, the probability of ending at p_{end} after m steps is the mean of the probabilities of ending at each p'_{end} after $m-1$ steps. This will be bounded by our bound on ending at any position after $m-1$ steps (noting that $m-1$ is even).

We first argue that the number of paths that a_1 can take ending in position $p_{end} = [p_1, ..., p_k]$ in round $r+m$ is upper bounded by the number of paths it can take ending at $[0, 0, ..., 0]$. We do this by describing an injection from paths ending at p_{end} to paths ending at $[0, 0, ..., 0]$.

Since m is even, the parity of $p_{end} = [p_1, ..., p_k]$ is even. We can thus arbitrarily pair up nonzero indices in this string. For any index pair (i, j), any path ending at p_{end} must take an odd number of steps in both the i and j directions. By flipping the last step in the path which is in either the i or j direction to the other direction, we obtain a path which makes an even number of steps in each direction. Further, this path is unique – given it, we can recover the path ending at p_{end} simply by again flipping the last step in either direction i or j to be in the other direction. In this way, but iterating through our arbitrary pairings, we can map any path ending at $[p_1, ..., p_k]$ to a unique path ending at $[0, ..., 0]$, giving us the injection.

We now bound the number of paths ending at $[0, ..., 0]$, which is identical to the number of ways m flips can be placed into k buckets, where each bucket has an even number of elements. This quantity is:

$$\sum_{\substack{s_1+...+s_k=m \\ (s_i \mod 2)\equiv 0}} \frac{m!}{s_1! \cdot \ldots \cdot s_k!}.$$

This value is equal to the coefficient of x^m in the exponential generating function

$$m!\left(1 + \frac{x^2}{2!} + \frac{x^4}{4!} + \ldots\right)^k = m!\left(\frac{e^x + e^{-x}}{2}\right)^k = \frac{m!}{2^k}\sum_{i=0}^{k}\binom{k}{i}e^{x(2i-k)}.$$

By differentiating m times, we find that the coefficient of x^m is:

$$\frac{1}{2^k}\sum_{i=0}^{k}\binom{k}{i}(2i-k)^m = \sum_{i=0}^{k}\left(\binom{k}{i}/2^k\right)\cdot(2i-k)^m.$$

This summation is exactly $\mathbb{E}\left[X^m\right]$, where X is a sum of k i.i.d. random variables each equal to 1 with probability $1/2$ and -1 otherwise. For any $c \in (0, 1]$, we can split the expectation:

$$\begin{aligned}\mathbb{E}\left[X^m\right] &= \mathbb{E}\left[X^m||X| \geq ck\right]\cdot\mathbb{P}[|X| \geq ck] + \mathbb{E}\left[X^m||X| \leq ck\right]\cdot\mathbb{P}[|X| \leq ck] \\ &\leq k^m\cdot\mathbb{P}[|X| \geq ck] + (ck)^m.\end{aligned}$$

To bound the return probability, we divide this count by the the total number of possible paths taken by a_1 m steps, k^m, giving an upper bound of:

$$\mathbb{P}[|X| \geq ck] + c^m.$$

By a Hoeffding bound, $\mathbb{P}[|X| \geq ck] \leq 2e^{-c^2k/2}$. If we set $c = \sqrt{\ln A/k} = \sqrt{\ln 2}$ then $\mathbb{P}[|X| \geq ck] \leq 1/\sqrt{A}$. So our final probability bound is:

$$\mathbb{P}[|X| \geq ck] + c^m \leq \frac{1}{\sqrt{A}} + (\sqrt{\ln 2})^m < \frac{1}{\sqrt{A}} + (9/10)^m,$$

yielding the lemma. Note that in the lemma the stated bound is $\frac{1}{\sqrt{A}} + (9/10)^{m-1}$ to account for odd m, as discussed. Also note that, by adjusting c, it is possible to trade off the terms in the above bound, giving stronger inverse dependence on A at the expense of slower exponential decay in m. □

Density Estimation Bound: Plugging Lemma 4.4.7 into Lemma 4.4.1, we have $B(t) = \sum_{m=0}^{t} \beta(m) \leq 10 + t/\sqrt{A}$. If we assume $t = O(\sqrt{A})$, this gives $\epsilon = O\left(\sqrt{\frac{\log(1/\delta)}{td}}\right)$ and so $t = O\left(\frac{\log(1/\delta)}{\epsilon^2 d}\right)$, matching independent sampling.

4.5 Independent-Sampling-Based Density Estimation

In this section we show that, if agents are not restricted to random walking, but can instead take arbitrary steps in each round, they can avoid collision correlations by splitting into 'stationary' and 'mobile' groups and counting collisions only between members of different groups. This allows them to essentially simulate independent sampling of grid locations to estimate density. This algorithm is not 'natural' in a biological sense, however it is easy to analyze and gives slightly better bounds than the random-walk-based approach on the grid (Theorem 4.3.1).

We give pseudocode in Algorithm 2. Recall that *position* is an ordered pair denoting an agent's (x, y) coordinates on the torus graph, and *count*(*position*) returns the number of other agents at the current position.

Algorithm 2 Independent-Sampling-Based Density Estimation

```
Each agent independently executes:
  Set $c := 0$ and with probability $1/2$, $state := walking$, else $state := stationary$.
  for $r = 1, ..., t$ do
      if $state := walking$ then
          $position := position + (0, 1)$          ▷ Deterministic walk step.
      end if
      $c := c + count(position)$                  ▷ Update collision count.
  end for
  $c := c \pmod t$
  return $\tilde{d} = \frac{2c}{t}$
```

Our main accuracy bound for the independent sampling algorithm is given below.

Theorem 4.5.1 (Independent Sampling Accuracy Bound)**.** *After running for t rounds, assuming $t < \sqrt{A}$ and $d \leq 1$, an agent executing Algorithm 2 returns $\tilde{d}$ such that, for any $\delta > 0$, with probability $\geq 1 - \delta$, $\tilde{d} \in [(1-\epsilon)d, (1+\epsilon)d]$ for $\epsilon = O\left(\sqrt{\frac{\log(1/\delta)}{td}}\right)$. In other words, for any $\epsilon, \delta \in (0, 1)$ if $t = \Theta\left(\frac{\log(1/\delta)}{d\epsilon^2}\right)$, $\tilde{d}$ is a $(1 \pm \epsilon)$ multiplicative estimate of d with probability $\geq 1 - \delta$.*

Proof. Our analysis is from the perspective of an agent with $state = walking$. By symmetry, the distribution of $\tilde{d}$ is identical for walking and stationary agents, so considering this case is sufficient.

Initially, assume that no two walking agents start in the same location. Given this assumption, we know that a walking agent *never collides with another walking agent* – by assumption they all start in different positions and update these positions identically in each round. In the written implementation, agents always step up, however any fixed pattern (e.g. a spiral) suffices.

Further, assume that agents do not execute the step of setting $c := c \pmod t$. This step will be used to handle walking agents which start at the same location, and will be analyzed at the end of the proof.

In t steps, a walking agent visits t unique squares (here we use the assumption that $t < \sqrt{A}$, the diameter of the grid). Each of the n other agents is located in this set of squares *and* stationary with probability $\frac{t}{2A}$. Further, each of these events is entirely independent from the rest, as the agents are positioned and choose their state independently. So, for a walking agent, c is just a sample of n independent random coin flips, each with success probability $\frac{t}{2A}$. Clearly, $\mathbb{E}[c] = n \cdot \frac{t}{2A} = \frac{td}{2}$ so

$\mathbb{E}[\tilde{d}] = \mathbb{E}\left[\frac{2c}{t}\right] = d$. Further, by a Chernoff bound, for any $\epsilon \in (0,1)$, the probability that $\tilde{d}$ is not a $(1 \pm \epsilon)$ multiplicative estimate of d is:

$$\delta = \mathbb{P}\left[|\tilde{d} - d| \geq \epsilon d\right] = \mathbb{P}\left[|c - \mathbb{E}[c]| \geq \epsilon\,\mathbb{E}[c]\right] \leq 2e^{-\epsilon^2\,\mathbb{E}[c]/3} \leq 2e^{-\epsilon^2 td/6}.$$

This gives: $\log(1/\delta) \geq \epsilon^2 td/6$ so $\epsilon = O\left(\sqrt{\frac{\log(1/\delta)}{td}}\right)$, yielding the result.

We now remove the assumption that no two walking agents start in the same location by considering the step where each agent sets $c := c \pmod t$ before returning $\tilde{d} = \frac{2c}{t}$. If an agent starts alone and is involved in $< t$ collisions, this operation has no effect – the above bound holds.

If a walking agent is involved in $< t$ 'true collisions' but starts in the same position as $w \geq 1$ other walking agents, the agents move in lockstep throughout the algorithm and are involved in $w \cdot t$ 'spurious collisions' (w in each round). Setting $c := c \pmod t$ exactly corrects for these spurious collisions and since c now only includes collisions with stationary agents, the bound above holds.

Finally, if an agent is involved in $\geq t$ true collisions, this modification cannot worsen their estimate. If $c \geq t$ and the agent does not set $c := c \pmod t$, they compute $\tilde{d} \geq \frac{2t}{t} \geq 2$. For $\epsilon < 1$, the agent fails since $d \leq 1$. So setting $c := c \pmod t$ can only increase the probability of success. □

4.6 Applications

We next discuss algorithmic applications of our ant-inspired density estimation algorithm (Algorithm 1) and the analysis techniques we develop.

4.6.1 Social Network Size Estimation

Random-walk-based density estimation is closely related to work on estimating the size of social networks and other massive graphs using random walks [KLSC14, KBM12, LL12, LW14]. In these applications, one does not have access to the full graph (so cannot exactly count the nodes), but can simulate random walks by following links between nodes [MMG$^+$07, GKBM09]. One approach is to run a single random walk and count repeat node visits [LL12, KBM12]. Alternatively, [KLSC14] proposes running multiple random walks and counting their collisions, which gives an estimate of the walk's density. Since the number of walks is known, this yields an estimate for network size.

This approach can be significantly more efficient since the dominant cost is typically in link queries to the network. With multiple, shorter random walks, this cost can be trivially distributed to multiple servers simulating walks independently. Visit information can then be aggregated and the collision count can be computed in a centralized manner.

Walks are first run for a 'burn-in period' so that their locations are distributed approximately by the stable distribution of the network. The walks are then halted, and the number of collisions in this final round are counted. The collision count gives an estimate of the walks' density. Since the number of walks is known, this yields an estimate for network size.

We show that ant-inspired algorithms can give runtime improvements over this method. After burn-in, instead of halting the walks immediately, we run them for *multiple rounds*, recording encounter rates as in Algorithm 1. This allows the use of fewer walks, decreasing total burn-in cost, and giving faster runtimes when mixing time is relatively slow, as is common in social network graphs [MYK10].

Random-Walk-Based Algorithm for Network Size Estimation

Consider an undirected, connected, non-bipartite graph $G = (V, E)$. Let S be the set of vertices of G that are 'known'. Initially, $S = \{v\}$ where v is a seed vertex. We can access G by looking up the neighborhood $\Gamma(v_i)$ of any vertex $v_i \in S$ and adding $\Gamma(v_i)$ to S.

To compute the network size $|V|$, we could scan S, looking up the neighbors of each vertex and adding them to the set. Repeating this process until no new nodes are added ensures that $S = V$ and we know the network size. However, this method requires $|V|$ neighborhood queries. The goal is to use significantly fewer queries using random-walk-based sampling.

A number of challenges are introduced by this application. While we can simulate many random walks on G, we can no longer assume these random walks start at randomly chosen nodes, as we do not have the ability to uniformly sample nodes from the network. Instead, we must allow the random walks to run for a burn-in phase of length proportional to the mixing time of G. After this phase, the walks are distributed approximately according to the stable distribution of G.

Further, in general G is not regular. In the stable distribution, a random walk is located at a vertex with probability proportional to its degree. Hence, collisions tend to occur more at higher degree vertices. To correct for this bias, we count a collision at vertex v_i with weight $1/\deg(v_i)$.

Our results depend on a natural generalization of re-collision probability. For any i, j, let $p(v_i, v_j, m)$ be the probability that an m-step random walk starting at v_i ends at v_j. Define:

$$\beta(m) \overset{\text{def}}{=} \frac{\max_{i,j} p(v_i, v_j, m)}{\deg(v_j)}.$$

Intuitively, $\beta(m)$ is the maximum m-step collision probability, weighted by degree since higher degree vertices are visited more in the stable distribution. Let $B(t) = \sum_{m=1}^{t} \beta(m)$. Note that this weighted $B(t)$ is trivially upper bounded by the unweighted measure used in Lemma 4.4.1.

For simplicity, we initially ignore burn-in and assume that our walks start distributed exactly by the stable distribution of G. A walk starts at vertex v_i with probability $p_i \overset{\text{def}}{=} \frac{\deg(v_i)}{\sum_i \deg(v_i)} = \frac{\deg(v_i)}{2|E|}$ and initial locations are independent. We also assume knowledge of the average degree $\overline{\deg} = 2|E|/|V|$. We later rigorously analyze burn-in and show to estimate $\overline{\deg}$, completing our analysis.

```
Algorithm 3 Random-Walk-Based Network Size Estimation
input: step count $t$, average degree $\overline{\deg}$, $n$ random starting locations $[w_1, ..., w_n]$ distributed independently according to the network's stable distribution
  $[c_1, ..., c_n] := [0, 0, ..., 0]$
  for $r = 1, ..., t$ do
      $\forall j$, set $w_j := randomElement(\Gamma(w_j))$ ▷ $\Gamma(w_j)$ denotes the neighborhood of $w_j$.
      $\forall j$, set $c_j := c_j + \frac{count(w_j)}{\deg(w_j)}$ ▷ $count(w_j)$ returns the number of other walkers currently at $w_j$.
  end for
  $C := \frac{\overline{\deg} \sum c_j}{n(n-1)t}$
  return $\tilde{A} = 1/C$
```

Note that there are many ways to implement the *count*$(\cdot)$ function used in Algorithm 3. One possibility is to simulate the random walks in parallel, recording their paths, and then to perform centralized post-processing to count collisions. As queries to the network are considered to dominate time cost, this collision counting step is relatively inexpensive.

Theorem 4.6.1. *If Algorithm 3 is run using n random walks for t steps, if $n^2 t =$*

$\Theta\left(\frac{B(t)\overline{\deg}+1}{\epsilon^2\delta}\cdot|V|\right)$, *then with probability at least* $1-\delta$, *it returns*

$$\tilde{A}\in[(1-\epsilon)|V|,(1+\epsilon)|V|]\,.$$

Analysis of Idealized Algorithm

We start with the analysis of Algorithm 3, which is given the average degree $\overline{\deg}$ as input and random walk starting locations distributed according to the network's stable distribution.

Throughout this section, we work directly with the weighted total collision count $C=\frac{\overline{\deg}\sum c_j}{n(n-1)t}$, showing that it is close to its expectation with high probability and hence giving the accuracy bound for $\tilde{A}$. As in the density estimation case, we start by showing that C is correct in expectation.

Lemma 4.6.2. $\mathbb{E}[C]=1/|V|$.

Proof. Let $c_j(r)$ be the number of collisions, weighted by inverse vertex degree, walk j expects to be involved in at round r. In each round all walks are at vertex v_i with probability $p_i=\frac{\deg(v_i)}{2|E|}$, so:

$$\mathbb{E}[c_j(r)]=\sum_{i=1}^{|V|}\left[\frac{\deg(v_i)}{2|E|}\cdot\frac{(n-1)\deg(v_i)}{2|E|}\cdot\frac{1}{\deg(v_i)}\right]=\frac{n-1}{4|E|^2}\sum_{i=1}^{|V|}\deg(v_i)=\frac{n-1}{2|E|}.$$

By linearity of expectation, $\mathbb{E}[c_j]=\frac{t(n-1)}{2|E|}$, $\mathbb{E}\left[\sum c_j\right]=\frac{tn(n-1)}{2|E|}$ and hence, $\mathbb{E}[C]=\frac{\overline{\deg}}{2|E|}=1/|V|$. □

We now show concentration of C about its expectation. Let $c_{i,j}$ be the weighted collision count between walks w_i and w_j where $i\neq j$. It is possible to follow the moment bound proof of Lemma 4.3.11 and bound all moments of $c_{i,j}$. However, there is no clear event we can condition on to ensure independence of all $c_{i,j}$'s. Hence, we cannot prove a bound analogous to Corollary 4.3.17 and employ the concentration result of Lemma 4.3.18.

Instead, we bound just second moment (the variance) of each $c_{i,j}$ and obtain our concentration results via Chebyshev's inequality as in Theorem 4.4.3. This leads to a linear rather than logarithmic dependence on the failure probability $1/\delta$. However, we note that we can simply perform $\log(1/\delta)$ estimates each with failure probability $1/3$ and return the median, which will be correct with probability $1-\delta$.

Lemma 4.6.3 (Degree Weighted Collision Variance Bound)**.** *For all $i, j \in [1, ..., n]$ with $i \neq j$, let $\bar{c}_{i,j} \overset{\text{def}}{=} c_{i,j} - \mathbb{E}[c_{i,j}]$. $\mathbb{E}\left[\bar{c}_{i,j}^2\right] = O\left(\frac{t(B(t)+|V|/|E|)}{|E|}\right)$.*

Proof. We can write $\mathbb{E}\left[\bar{c}_{i,j}^2\right] = \mathbb{E}\left[c_{i,j}^2\right] - \left(\mathbb{E}\left[c_{i,j}\right]\right)^2 \leq \mathbb{E}\left[c_{i,j}\right]^2$. We can then split $c_{i,j}$ over rounds to give:

$$\mathbb{E}\left[\bar{c}_{i,j}^2\right] \leq \mathbb{E}\left[\left(\sum_{r=1}^{t} c_{i,j}(r)\right)^2\right] = \sum_{r=1}^{t} \mathbb{E}\left[c_{i,j}(r)^2\right] + 2\sum_{r=1}^{t-1}\sum_{r'=r+1}^{t} \mathbb{E}\left[c_{i,j}(r)c_{i,j}(r')\right].$$

Since the walks are in the stable distribution, and hence located at v_i in each round with probability $\frac{\deg(v_i)}{2|E|}$, we have the weighted collision $c_{i,j}(r) = \frac{1}{\deg(v_i)}$ with probability $\frac{\deg(v_i)^2}{(2|E|)^2}$. We thus have $\mathbb{E}\left[c_{i,j}(r)^2\right] = \sum_{i=1}^{|V|}\left(\frac{\deg(v_i)^2}{(2|E|)^2} \cdot \frac{1}{\deg(v_i)^2}\right)$. $\mathbb{E}\left[c_{i,j}(r)c_{i,j}(r')\right]$ can be computed similarly by summing over all pairs of vertices $\frac{1}{\deg(v)\deg(u)}$ times the probability that the agents collide at vertex v in round r and then again at vertex u in round r'. Overall this gives:

$$\begin{aligned}\mathbb{E}\left[\bar{c}_{i,j}^2\right] &\leq t\sum_{i=1}^{|V|}\left(\frac{\deg(v_i)^2}{(2|E|)^2} \cdot \frac{1}{\deg(v_i)^2}\right) \\ &\quad + 2\sum_{r=1}^{t-1}\sum_{r'=r+1}^{t}\left(\sum_{i=1}^{|V|}\left(\frac{\deg(v_i)^2}{(2|E|)^2} \cdot \frac{1}{\deg(v_i)} \cdot \sum_{j=1}^{|V|}\frac{p(v_i, v_j, r-r')^2}{\deg(v_j)}\right)\right) \\ &\leq \frac{t|V|}{4|E|^2} + 2t\sum_{m=1}^{t-1}\left(\sum_{i=1}^{|V|}\left(\frac{\deg(v_i)}{(2|E|)^2} \cdot \beta(m)\sum_{j=1}^{|V|} p(v_i, v_j, m)\right)\right)\end{aligned}$$

where in the last step we write $r - r' = m$ and use the fact that $\beta(m) \overset{\text{def}}{=} \frac{\max_{i,j} p(v_i,v_j,m)}{\deg(v_j)}$. We have $\sum_{j=1}^{|V|} p(v_i, v_j, m) = 1$ and so can simplify the above as:

$$\begin{aligned}\mathbb{E}\left[\bar{c}_{i,j}^2\right] &\leq \frac{t|V|}{4|E|^2} + 2t\sum_{m=1}^{t-1}\frac{\sum_{i=1}^{|V|}\deg(v_i)}{(2|E|)^2} \cdot \beta(m) \\ &= \frac{t|V|}{4|E|^2} + 2t\sum_{m=1}^{t-1}\frac{\beta(m)}{2|E|} = O\left(\frac{t(B(t)+|V|/|E|)}{|E|}\right).\end{aligned}$$

□

Lemma 4.6.4 (Total Collision Variance Bound)**.** *Let $\overline{C} = \frac{\overline{\deg}\sum_j \bar{c}_j}{n(n-1)t}$.*

$$\mathbb{E}\left[\bar{C}^2\right] = O\left(\frac{1}{n^2 t} \cdot \frac{B(t)|E| + |V|}{|V|^2}\right).$$

Proof. $\sum_{j=1}^{n} \bar{c}_j = \sum_{i,j \in [1,\ldots,n], i \neq j} \bar{c}_{i,j}$. We closely follow the variance calculation in [KLSC14]:

$$\begin{aligned}\mathbb{E}\left[\left(\sum_{i,j\in[1,\ldots,n],i\neq j} \bar{c}_{i,j}\right)^2\right] &= \sum_{i,j\in[1,\ldots,n],i\neq j} \left[\sum_{i',j'\in[1,\ldots,n],i\neq j} \bar{c}_{i,j} \cdot \bar{c}_{i',j'}\right] \\ &= 2\binom{n}{2} \mathbb{E}\left[\bar{c}_{i,j}^2\right] + 4!\binom{n}{4} \mathbb{E}[\bar{c}_{i,j}]^2 + 2\cdot 3!\binom{n}{3} \mathbb{E}[\bar{c}_{i,j}\bar{c}_{i,k}].\end{aligned}$$

The first term corresponds to the cases when $i = i'$ and $j = j'$. The second corresponds to $i \neq i'$ and $j \neq j'$, in which case $\bar{c}_{i,j}$ and $\bar{c}_{i',j'}$ are independent and identically distributed. The $4!\binom{n}{4}$ multiplier is the number of ways to choose an ordered set of four distinct indices. The last term corresponds to all cases when either $i = i'$ or $j = j'$. There are $3!\binom{n}{3}$ ways to choose an ordered set of three distinct indices, multiplied by two to account for the repeated index being in either the first or second position. Using $\mathbb{E}[\bar{c}_{i,j]} = 0$ and the bound on $\mathbb{E}\left[\bar{c}_{i,j}^2\right]$ from Lemma 4.6.3:

$$\mathbb{E}\left[\left(\sum_{i,j\in[1,\ldots,n],i\neq j} \bar{c}_{i,j}\right)^2\right] = O\left(\frac{n^2 t(B(t) + |V|/|E|)}{|E|}\right) + 0 + 2\cdot 3!\binom{n}{3} \mathbb{E}[\bar{c}_{i,j}\bar{c}_{i,k}]. \tag{4.10}$$

When $j \neq k$, $\bar{c}_{i,j}$ and $\bar{c}_{i,k}$ are independent and identically distributed conditioned on the path that walk w_i traverses (this is similar to the independence used to prove Corollary 4.3.17). Let Ψ_i be the t-step path chosen by w_i.

$$\begin{aligned}\mathbb{E}\left[\bar{c}_{i,j}\bar{c}_{i,k}\right] &= \sum_{\psi_i} \mathbb{P}\left[\Psi_i = \psi_i\right] \cdot \mathbb{E}\left[\bar{c}_{i,j}|\Psi_i = \psi_i\right] \cdot \mathbb{E}\left[\bar{c}_{i,k}|\Psi_i = \psi_i\right] \\ &= \sum_{\psi_i} \mathbb{P}\left[\Psi_i = \psi_i\right] \cdot \mathbb{E}\left[\bar{c}_{i,j}|\Psi_i = \psi_i\right]^2 \\ &= \sum_{\psi_i} \mathbb{P}\left[\Psi_i = \psi_i\right] \cdot \left(\mathbb{E}\left[c_{i,j}|\Psi_i = \psi_i\right] - \mathbb{E}\left[c_{i,j}\right]\right)^2. \end{aligned} \tag{4.11}$$

$\mathbb{E}\left[c_{i,j}|\Psi_i = \psi_i\right] = \sum_{r=1}^{t} \frac{\deg(\psi_i(r))}{2|E|} \cdot \frac{1}{\deg(\psi_i(r))} = \frac{t}{2|E|} = \mathbb{E}\left[c_{i,j}\right]$. That is, the expected number of collisions is identical for every path of w_i. Plugging into (4.11),

$$\mathbb{E}\left[\bar{c}_{i,j}\bar{c}_{i,k}\right] = 0.$$

So finally, plugging back into equation (4.10),

$$\mathbb{E}\left[\left(\sum_{i,j\in[1,\ldots,n],i\neq j} \bar{c}_{i,j}\right)^2\right] = O\left(\frac{n^2t(B(t)+|V|/|E|)}{|E|}\right)$$

and thus:

$$\begin{aligned}\mathbb{E}\left[\overline{C}^2\right] &= O\left(\frac{n^2t(B(t)+|V|/|E|)}{|E|}\cdot\left(\frac{\overline{\deg}}{n(n-1)t}\right)^2\right)\\ &= O\left(\frac{1}{n^2t}\cdot\frac{(B(t)+|V|/|E|)\cdot|E|}{|V|^2}\right)\\ &= O\left(\frac{1}{n^2t}\cdot\frac{B(t)|E|+|V|}{|V|^2}\right).\end{aligned}$$

□

With this variance bound in place, we can finally prove Theorem 4.6.1.

Proof of Theorem 4.6.1. Note that $\bar{C} = C - \mathbb{E}[C]$ and by Lemma 4.6.2, $\mathbb{E}[C] = 1/|V|$. By Chebyshev's inequality Lemma 4.6.4 gives:

$$\mathbb{P}\left[|C-\mathbb{E}[C]| \geq \epsilon\,\mathbb{E}[C]\right] \leq \frac{1}{\epsilon^2 n^2 t}\cdot(B(t)|E|+|V|).$$

Rearranging gives us that, in order to have $C \in \left[\frac{1-\epsilon}{|V|}, \frac{1+\epsilon}{|V|}\right]$ with probability δ, we must have:

$$n^2t = \Theta\left(\frac{B(t)|E|+|V|}{\epsilon^2\delta}\right).$$

Since $\tilde{A} = 1/C$, if $C \in \left[\frac{1-\epsilon}{|V|}, \frac{1+\epsilon}{|V|}\right]$ then $\tilde{A} \in \left[\frac{|V|}{1+\epsilon}, \frac{|V|}{1-\epsilon}\right] \subseteq [(1-2\epsilon)|V|, (1+2\epsilon)|V|]$ as long as $\epsilon < 1/2$. This gives the theorem after adjusting constants on ϵ and recalling that $\overline{\deg} = |E|/|V|$. □

Estimating The Average Degree

We now show how to estimate the value of $\overline{\deg}$ used in Algorithm 3. Specifically, we need a $(1\pm\epsilon)$ approximation to $\frac{1}{\overline{\deg}}$. If we then substitute this into the formula $\tilde{A} = \frac{\sum_j c_j}{\overline{\deg}\cdot n(n-1)t}$, we still have a $(1\pm O(\epsilon))$ approximation to the true network size. We use the algorithm and analysis of [KLSC14], which gives a simple approximation via inverse degree sampling.

Algorithm 4 Average Degree Estimation

```
input: $n$ random starting locations $[w_1, ..., w_n]$ distributed independently according to the network's stable distribution.
    $\forall j$, set $d_j := \frac{1}{\deg(w_j)}$                                ▷ Sampling
    return $D := \frac{\sum d_j}{n}$
```

Theorem 4.6.5 (Average Degree Estimation)**.** *If* $n = \Theta\left(\frac{1}{\epsilon^2\delta} \cdot \frac{\overline{\deg}}{\deg_{\text{min}}}\right)$*, Algorithm 4 returns D such that, with probability at least $1-\delta$, $D \in \left[\frac{1-\epsilon}{\overline{\deg}}, \frac{1+\epsilon}{\overline{\deg}}\right]$.*

Proof. Using that in the stable distribution a walk is at vertex v_i with probability $\frac{\deg(v_i)}{2|E|}$ we have:

$$\mathbb{E}[D] = \frac{1}{n}\sum_{j=1}^{n}\mathbb{E}[d_j] = \frac{1}{n}\cdot n \cdot \sum_{i=1}^{|V|}\left(\frac{\deg(v_i)}{2|E|}\cdot\frac{1}{\deg(v_i)}\right) = \frac{|V|}{2|E|} = \frac{1}{\overline{\deg}}.$$

For each d_j let $\bar{d}_j = d_j\,\mathbb{E}[d_j]$. We have $\mathbb{E}[\bar{d}_j^2] = \mathbb{E}[d_j^2] - \mathbb{E}[d_j]^2 \leq \mathbb{E}[d_j^2]$. We can explicitly compute this expectation as:

$$\mathbb{E}[d_j^2] = \sum_{i=1}^{|V|}\frac{\deg(v_i)}{2|E|}\frac{1}{\deg(v_i)^2} \leq \frac{|V|}{2|E|\deg_{\text{min}}} = \frac{1}{\deg_{\text{min}}}\cdot\frac{1}{\overline{\deg}}.$$

Additionally, since each d_j is independent and identically distributed, and since $\bar{D} = \frac{1}{n}\sum d_j$, letting $\bar{D} = D - \mathbb{E}[D]$,

$$\mathbb{E}\left[\bar{D}^2\right] = \frac{1}{n}\mathbb{E}[\bar{d}_j^2] \leq \frac{1}{n}\mathbb{E}[d_j^2] = \frac{1}{n\deg_{\text{min}}}\cdot\frac{1}{\overline{\deg}}$$

Applying Chebyshev's inequality and the fact that $\mathbb{E}[D] = \frac{1}{\overline{\deg}}$:

$$\mathbb{P}\left[|D - \mathbb{E}[D]| \leq \frac{\epsilon}{\overline{\deg}}\right] \leq \frac{\overline{\deg}}{\epsilon^2 n \deg_{min}}.$$

Rearranging, to succeed with probability $\geq 1-\delta$ it suffices to set $n = \Theta\left(\frac{1}{\epsilon^2\delta}\cdot\frac{\overline{\deg}}{\deg_{\text{min}}}\right)$. □

Handling Burn-In Error

Finally, we remove our assumption that walks start distributed exactly according to the network's stable distribution, rigorously bounding the length of burn-in required before running Algorithm 3.

Let $\mathcal{D}^* \in \mathbb{R}^{|V|^n}$ be a vector representing the true stable distribution of n random walks on G and $\mathcal{D}_t \in \mathbb{R}^{|V|^n}$ be a vector representing the distribution of the walks after running for t burn-in steps. Specifically, each walk $w_1, ..., w_n$ is initialized at a single seed vertex v. For t rounds we then update the location of each walk independently by moving to a randomly chosen neighbor. Both vectors are probability distributions: they have all entries in $[0, 1]$ and $\|\mathcal{D}^*\|_1 = \|\mathcal{D}\|_1 = 1$.

Let $\Delta = \mathcal{D}^* - \mathcal{D}_t$ and assume that $\|\Delta\|_1 \leq \delta$. We can consider two equivalent algorithms: draw an initial set of locations $W = w_1, ..., w_n$ from $\mathcal{D}^*$, run Algorithm 3, and then artificially fail with probability $\max\{0, \Delta(W)\}$. Alternatively, draw $W = w_1, ..., w_n$ from $\mathcal{D}_t$, run Algorithm 3, and then artificially fail with probability $\max\{0, -\Delta(W)\}$. These algorithms are clearly equivalent. The first obtains a good estimator with probability $1-2\delta$: probability δ that Algorithm 3 fails when initialized via the stable distribution $\mathcal{D}^*$ by Theorem 4.6.1 plus an artificial failure probability of $\leq \|\Delta\|_1 \leq \delta$. The second then clearly also fails with probability 2δ. This can only be higher than if we did not perform the artificial failure after running Algorithm 3. Therefore, running Algorithm 3 with a set of random walks initially distributed according to $\mathcal{D}_t$ yields success probability $\geq 1 - 2\delta$.

How long must the burn-in period be to ensure $\|\mathcal{D}^* - \mathcal{D}_t\|_1 \leq \delta$? Let $\mathbf{W}$ be the random walk matrix of G. Let $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_A$ be the eigenvalues of $\mathbf{W}$ and $\lambda = \max\{|\lambda_2|, |\lambda_{|V|}|\}$. Let $\mathcal{C}_t \in \mathbb{R}^{|V|}$ denote the location distribution for a single random walk after burn-in and $\mathcal{C}^* \in \mathbb{R}^{|V|}$ denote the stable distribution of a single random walk. If we have, for all i, $|\mathcal{C}_t(v_i) - \mathcal{C}^*(v_i)| \leq \delta/n \cdot \mathcal{C}^*(v_i)$ then for any W:

$$\begin{aligned}
|\mathcal{D}_t(W) - \mathcal{D}^*(W)| &= \left| \prod_{i=1}^{n} \mathcal{C}_t(w_i) - \prod_{i=1}^{n} \mathcal{C}^*(w_i) \right| \\
&\leq \prod_{i=1}^{n} (\mathcal{C}^*(w_i) + \delta/n \cdot \mathcal{C}^*(w_i)) - \prod_{i=1}^{n} \mathcal{C}^*(w_i) \\
&< \mathcal{D}^*(W) \sum_{i=1}^{n} \binom{n}{i} (\delta/n)^i \leq \mathcal{D}^*(W) \sum_{i=1}^{n} \delta^i \leq 2\delta \cdot \mathcal{D}^*(W),
\end{aligned}$$

as long as $\delta < 1/2$. This multiplicative bound gives $\|\mathcal{D}^* - \mathcal{D}_t\|_1 \leq 2\delta$. By standard mixing time bounds ([Lov93], Theorem 5.1), $|\mathcal{C}_t(v_i) - \mathcal{C}^*(v_i)| \leq \frac{\delta}{n|E|} \cdot \mathcal{C}^*(v_i)$ for all i after $M = O\left(\frac{\log(n|E|/\delta)}{1-\lambda}\right) = O\left(\frac{\log(|E|/\delta)}{1-\lambda}\right)$ burn-in steps (since $n < |E|$ or else we could have scanned the full graph.)

Overall Runtime and Comparison to Previous Work

Let $M = O\left(\frac{\log(|E|/\delta)}{1-\lambda}\right)$ denote the burn-in time required before running Algorithm 3. In order to obtain a $(1 \pm \epsilon)$ estimate of network size with probability $1-\delta$ we must run n random walks for $M+t$ steps, making $n(M+t)$ link queries, where by Theorems 4.6.1 and 4.6.5:

$$n = \Theta\left(\max\left\{\frac{\overline{\deg}}{\deg_{\text{min}}\epsilon^2\delta}, \sqrt{\frac{|V| \cdot (B(t)\overline{\deg}+1)}{t \cdot \epsilon^2 \delta}}\right\}\right). \tag{4.12}$$

Typically, the second term dominates since $\overline{\deg} << |V|$. Hence, by increasing t, we are able to use fewer random walks, significantly decreasing the number of link queries if M is large.

[KLSC14] uses a different approach, halting random walks and counting collisions immediately after burn-in. For reasonable node degrees they require $n = \Theta\left(\frac{|V| \cdot \overline{\deg}}{\epsilon^2\delta \cdot \sqrt{\sum \deg(v_i)^2}}\right)$. Assuming that $\sqrt{\sum \deg(v_i)^2} < n$, and setting $t = 1$, this is somewhat smaller than our bound as $\sum \deg(v_i)^2 \geq |V| \cdot \overline{\deg}$. However, (4.12) gives an important tradeoff – by increasing t we can increase the number of steps in our random walks, decreasing the total number of walks.

As an illustrative example, consider a k-dimensional torus graph for $k \geq 3$ (for $k = 2$ mixing time is $\Theta(|V|)$ so we might as well census the full graph). The burn-in mixing time required for Algorithm 3 is $M = \Theta(\log(|V|/\delta)|V|^{2/k})$. All nodes have degree $2k$, and using the bounds above, to obtain a $(1 \pm \epsilon)$ estimate of $|V|$, the algorithm of [KLSC14] requires

$$M \cdot n = \Theta\left(\frac{\log(|V|/\delta)}{\epsilon\sqrt{d}} \cdot |V|^{2/k+1/2}\right)$$

link queries. In contrast, assuming $|V|$ is large, we require $n = \Theta\left(\sqrt{\frac{|V|}{t \cdot \epsilon^2\delta}}\right)$ since by Lemma 4.4.4, $B(t) = O(1/k)$ and $\overline{\deg} = \deg_{\text{min}} = k$. If we set $t = \Theta(M)$, the total number of link queries needed is

$$n(M+t) = O\left(\frac{\sqrt{\log(|V|/\delta)}}{\epsilon\sqrt{d}} \cdot |V|^{(k+1)/2k}\right).$$

This beats [KLSC14] by improving dependence on $|V|$ and the logarithmic burn-in term. Ignoring error dependencies, if $k = 3$, [KLSC14] requires $\Theta(n^{7/6})$ queries which is more expensive than fully censusing the graph. We require $O(n^{2/3})$ queries, which is sublinear in the graph size.

We leave open comparing our bounds with those of [KLSC14] on more natural classes of graphs. It would be interesting to determine typical values of $B(t)$ in real work networks or popular graph models, such as preferential attachment models and others with power-law degree distributions.

4.6.2 Distributed Density Estimation by Robot Swarms

Algorithm 1 can be directly applied as a simple and robust density estimation algorithm for robot swarms moving on a two-dimensional plane modeled as a grid. Additionally, the algorithm can be used to estimate the frequency of certain properties within the swarm. Let d be the overall population density and d_P be the density of agents with some property P. Let $f_P = d_P/d$ be the relative frequency of P.

Assuming that agents with property P are distributed uniformly in population and that agents can detect this property (through direct communication or some other signal), then they can separately track encounters with these agents. They can compute an estimate $\tilde{d}$ of d and $\tilde{d}_P$ of d_P. By Theorem 4.3.1, after running for $t = \Theta\left(\frac{\log(1/\delta)[\log\log(1/\delta)+\log(1/d\epsilon)]^2}{d_P\epsilon^2}\right)$ steps, with probability $1-2\delta$,

$$\tilde{d}_P/\tilde{d} \in \left[\left(\frac{1-\epsilon}{1+\epsilon}\right) f_P, \left(\frac{1+\epsilon}{1-\epsilon}\right) f_P\right] = [(1-O(\epsilon))f_P, (1+O(\epsilon))f_P]$$

for small ϵ.

In an ant colony, properties may include whether or not an ant has recently completed a successful foraging trip [Gor99], or if an ant is a nestmate or enemy [Ada90]. In a robotics setting, properties may include whether a robot is part of a certain task group, whether it has completed a certain task, or whether it has detected a certain event or environmental property.

4.7 Discussion and Future Work

We have presented a theoretical analysis of random-walk-based density estimation by agents on a two-dimensional torus graph. We have also presented applications of our techniques to density estimation on other regular graph topologies and to the problems of social network size estimation and density estimation on robot swarms. Our work leaves open a number of open questions which we discuss below.

4.7.1 Extensions to Our Model

We feel that our simple model of density estimation on the two-dimensional torus (Section 4.2) well reflects the behavior of ants estimating density via collision rates while moving around a two-dimensional surface. However, extending our results to more realistic models would be a very interesting direction.

We believe that it is important to consider a model in which agents are not positioned uniformly at random on the torus. Without the uniform placement assumption, solving the global density estimation problem that we have defined may become difficult. If most agents are placed in a very small portion of the torus, any other agent initially located far away from these agents must traverse a large portion of the torus to find them with good probability, and hence, to estimate the global population density accurately.

There are several ways to overcome this difficulty. First, given some distribution of the agents over the torus, it may be possible to give bounds parameterized by the distance from this distribution to the uniform distribution. If the distribution is close to uniform, random-walk-based estimation should do a good job estimating density. If it is very far from uniform (e.g., in the example above, where many agents are concentrated in a small area), global density estimation will become more difficult.

Alternatively, as discussed in Section 4.2.2, it would be very interesting to formally define a *local density estimation problem*, which takes an agent's initial location into account when defining the density which they aim to estimate. In such a problem, agents located in more densely populated areas of the torus will return higher local density estimates.

Another interesting direction is to modify our assumption that agents move via random walk, considering more complex models based on empirical studies of ant movement [GPT93, NTD05, BFKN18]. It may be interesting to study a model in which agents sense and sometimes avoid collisions, or in which they move away from previously encountered ants. It may also be interesting to consider random-walk-based models, but with asynchronous movement, or continuous movement along a surface. Empirically, while there is some work directly testing the assumption of random movement by considering re-collision rates [BFKN18], providing further evidence of how closely our model predicts re-collision probabilities and, in turn, density estimation accuracy, would be very interesting.

Finally, it would be valuable to explore the robustness of random-walk-based density estimation to noise and other perturbations. One possibility is to model noisy collision detection, in which each collision is only detected with some probability, or in which spurious collisions may occasionally be detected. We may also model noise in ant behavior. For example, each agent may not move via pure random walk, but via some perturbed behavior which assigns nonuniform probabilities to the steps $\{(0,1),(0,-1),(1,0),(-1,0),(0,0)\}$.

4.7.2 Biological Applications

As discussed in Section 4.1, density estimation is used as a subroutine in many ant behaviors such as quorum sensing [Pra05] in house-hunting, task allocation [Gor99, SHG06], and appraisal of enemy colony strength [Ada90]. Modeling these behaviors theoretically, and studying how our approximate density estimation results can be composed with higher level algorithms is a very interesting direction.

In our own work, we have considered the use of density estimation in the house-hunting process in *Temnothorax* ants [Rad17, RML17], demonstrating that approximate density estimation , where the density estimate is correct in expectation and within a $(1 \pm \epsilon)$ factor of the true density with high probability, suffices for efficient decision making in the house-hunting process. It would be interesting, for example, to prove similar results for task-allocation, where density estimation may be used to approximate the number of workers currently performing a given task.

Density estimation behavior may also be used in other species. For example, there is evidence that higher work density stimulates certain reproductive behaviors in honeybee colonies [SKP17]. Studying theoretically how bees estimate and respond to increased density, and how this behavior compares to ant colony behavior would be valuable.

Finally, we note that the accuracy bound of Theorem 4.3.1 depends inversely on the density d, and so becomes large when d is small. In many of the above biological applications, such as in quorum sensing for decision making in ant colonies, agents only need to detect when d is above some fixed threshold. In this case, better bounds, where t depends not on the true density, but just on this detection threshold, may be possible. Additionally, it may be interesting to understand how multiple agents with different density estimates can cooperate to learn if a density threshold has been reached, with more accuracy than if just a single agent were attempting to detect such a threshold.

4.7.3 Algorithmic Applications

We conclude by discussing algorithmic applications of our analysis techniques, extending the results presented in Section 4.6.

Random-Walk-Based Sensor Network Sampling

We believe our moment bounds for a single random walk (Corollaries 4.3.15 and 4.3.16) can be applied to random-walk-based distributed algorithms for sensor network sampling. Random-walk-based sensor network sampling [LB07, AB04] is a technique in which a query message (a 'token') is initially sent by a base station to some sensor. The token is relayed randomly between sensors, which are connected via a grid communication network, and its value is updated appropriately at each step to give an answer to the query. This scheme is robust and efficient - it easily adapts to node failures and does not require setting up or storing spanning tree communication structures.

Random-walk-based sampling could be used, for example, to estimate the percentage of sensors that have recorded a specific condition, or the average value of some measurement at each sensor. However, as in density estimation, unless an effort is made to record which sensors have been previously visited, additional error may be added due to repeat visits. Recording previous visits introduces computational burden – either the token message size must increase or nodes themselves must remember which tokens they have seen. We are hopeful that our moment bounds can be used to show that this is unnecessary – due to strong local mixing, the number of repeat sensor visits will be low, and the performance reduction limited.

We remark that estimating the percentage of sensors in a network or the density of robots in a swarm with a property that is uniformly distributed is a special case of a more general *data aggregation* problem: each agent or sensor holds a value v_i drawn independently from some distribution $\mathcal{D}$. The goal is to estimate some statistic of $\mathcal{D}$, such as its expectation. In the case of density estimation, v_i is simply an indicator random variable which is 1 with probability d and 0 otherwise. Extending our results to more general data aggregation problems and showing that random walk sampling matches independent sampling in some cases is an interesting future direction.

Size Estimation of Realistic Networks

We leave open studying the effectiveness of the algorithm for social network size estimation presented in Section 4.6.1 in real world networks, or on popular random graph models for social networks [NWS02]. It may also be interesting to give bounds for the algorithm in general graphs, parameterized by the global mixing time, rather than the m-step recollision probability $\beta(m)$. While such bounds may give a coarser characterization of the algorithm's performance, they could be used to compare again worst case bounds for related random-walk-based network size estimation approaches parameterized by the mixing time [BHOP18].

Beyond Encounter Rate

In social network size estimation, robot swarm density estimation, and sensor network sampling, it is possible to leverage more information than just the random walk encounter rate. For example, a size estimation algorithm can store each agent's full t-step path, and count the number of intersections between these paths. A robot swarm algorithm may assign ids to each agent and use them to identify repeat collisions. It would be valuable to understand if these strategies can be used to improve our bounds, or if they do not give significant advantages.

Other Potential Applications

Finally, there are many potential applications of random-walk-based density estimation and sampling that we have not yet considered. For example, density estimation may be employed as a subroutine in swarm robot coverage and exploration routines, which aim to explore an unknown environment, or survey a known environment, as quickly as possible [BMF$^+$00, BS02]. It may be interesting to use density estimation to detect regions with high robot density, and to then spread out this density to more efficiently distribute exploration. Similar techniques may be interesting in robot formation problems in which the goal is the spread a swarm of robots or sensors regularly across a surface (or according to some specified distribution) using a distributed algorithm [GCD$^+$03, CMKB04, GLMN09].

Chapter 5

Computation in Spiking Neural Networks

In this chapter, we study biological neural networks from an algorithmic perspective, focusing on understanding tradeoffs between computation time and network complexity. We use a biologically plausible yet simplified neural computational model. Our goal is to abstract real neural networks in a way that, while not capturing all interesting features, preserves high-level behavior and allows us to make biologically relevant conclusions. Towards this goal, we consider the implementation of algorithmic primitives in a simple yet biologically plausible model of *stochastic spiking neural networks*.

We show how the stochastic behavior of neurons in this model can be leveraged to solve a basic *symmetry-breaking task* in which we are given neurons with identical firing rates and want to select a distinguished one. In computational neuroscience, this is known as the winner-take-all (WTA) problem, and it is believed to serve as a basic building block in many tasks, e.g., learning, pattern recognition, and clustering. We provide efficient constructions of WTA circuits in our stochastic spiking neural network model, as well as lower bounds in terms of the number of auxiliary neurons required to drive convergence to WTA in a given number of steps. These lower bounds demonstrate that our constructions are near-optimal in some cases.

This chapter covers work originally published in [LMP17a]. In related work, [LMP17b, LMP17c] we study simple compression and similarity testing tasks in the same model of neural computation.

5.1 Background and Introduction to Results

Neural networks are studied in a number of academic communities from a wide range of perspectives. Significant work in computational neuroscience focuses on developing somewhat realistic mathematical models for these networks and generally studying their capacity to process information [Izh04, Tra09]. On the more theoretical side, a variety of artificial network models such as perceptron and sigmoidal networks, Hopfield networks, and Boltzmann machines have been developed. These models are tractable to theoretical analysis and studied in the context of their computational power, and applications to general function approximation, classification, and memory storage [HSW89, MSS91, SS95, Maa97]. In practical machine learning, biological fidelity and often theoretical tractability are put aside, and researchers study how neural-like networks and learning rules can be used to efficiently represent and learn complex concepts [Hay09, LBH15].

In contrast to the common approach in computational neuroscience and machine learning, in our work we focus not on general computation ability or broad learning tasks, but on specific algorithmic implementation and analysis. We define a model of neural computation along with algorithmic problems that seem to be an important building blocks for higher level processing and learning tasks. We then design neural networks in our model that solve these problems, rigorously analyzing the complexity of our solutions in terms of asymptotic runtime and network size bounds. We hope that this new paradigm will provide new insights about computational tradeoffs, the power of randomness, and the role of noise in biological systems.

While focusing on somewhat different questions, our line of work is inspired by (1) work on the computational power of spiking neural networks, most notably by Maass et al. [Maa97, Maa99, Maa00] and (2) the work of Les Valiant [Val00a, Val00b, Val05], who defined the *neuroidal model* of computation and investigated implementations of basic learning modules within this model.

5.1.1 Spiking Neural Networks

We consider a model of *spiking neural networks* (SNNs) [Maa96, Maa97, GK02, Izh04, HJM13], defined formally in Section 5.2, in which neurons fire in discrete pulses, in response to a sufficiently high membrane potential. This potential is induced by spikes from neighboring neurons, which can have either an excitatory or inhibitory effect (increasing or decreasing the potential). Our model is *stochastic* – each neuron functions as a probabilistic threshold unit, spiking with probability given by applying a sigmoid function to the membrane potential. In this respect, our networks are similar to the popular Boltzmann machine [AHS85], with the important distinction that synaptic weights are not required to be symmetric and, as observed in nature, neurons are either strictly inhibitory (all outgoing edge weights are negative) or excitatory. Additionally, in this thesis, we focus on networks with fixed edge weights. The literature on Boltzmann machines tends to focus on learning, in which edge weights are adjusted iteratively until the network converges to some desired distribution on firing patterns. While a rich literature focuses on deterministic threshold circuits [MP69, HT$^+$86] we employ a stochastic model as it is widely accepted that neural computation is inherently stochastic [AS94, SN94, FSW08], and that while this can lead to a number of challenges, it also affords significant computational advantages [Maa14].

5.1.2 The Winner-Take-All Problem

In this chapter, we focus on the Winner-Take-All (WTA) problem, which is one of the most studied problems in computational neuroscience. A WTA network has n input neurons, n corresponding outputs, and a set of auxiliary neurons that facilitate computation. The goal is to pick a 'winning' input – that is, the network should produce a single firing output which corresponds to a firing input. Often the winning input is the one with the highest firing rate, in which case WTA serves as a neural max function. We focus on the case when all inputs have the same or similar firing rates, in which case WTA serves as a leader election unit. A formal definition of the WTA problem is given in Section 5.2.6.

WTA is widely applicable, including in circuits that implement visual attention via WTA competition between groups of neurons that process different input classes [KU87, LIKB99, IK01]. It is also the foundation of competitive learning [Now89, KK94, GL09], in which classifiers compete to respond to specific input types. More broadly, WTA is known to be a powerful computational primitive [Maa99, Maa00] – a network equipped with WTA units can perform some tasks significantly more efficiently than with just linear threshold neurons (McCulloch-Pitts neurons or perceptrons).

Due to its importance, there has been significant work on WTA, including in biologically plausible spiking networks [LRMM88, YG89, Tho90, CGL92, WS03, OL06, ODL09, ASNN$^+$15]. This work is extremely diverse – while mathematical analysis is typically given, different papers show different guarantees and apply varying levels of rigor. To the best of our knowledge, prior to our work, no asymptotic time bounds (e.g., as a function of the number of inputs n) for solving WTA in spiking neural networks have been established.[1] Additionally, previous analysis often requires a specific initial network state to show convergence and does not show that the network is self-stabilizing and converges from an arbitrary starting state, as is necessary in a biological system.

5.1.3 Our Contributions

We explore the tradeoff between the number of auxiliary neurons used in a WTA network (i.e., the complexity of the network) and the time required to select a winning output (to converge to a WTA state).

Network Constructions and Runtime Bounds

One the upper bound side, in Section 5.3 we describe, for any input size n and failure probability $\delta > 0$, a family of networks using just two auxiliary inhibitory neurons which solve the WTA problem in $O(\log n)$ steps in expectation, and with probability $\geq 1 - \delta$ in $O(\log n \cdot \log(1/\delta))$ steps (see Theorems 5.3.2 and 5.3.3).

Our two-inhibitor construction is based on a simple random competition idea. Outputs that fire in response to stimulation from their firing inputs excite two inhibitors, which, in turn, inhibit all the outputs. When more than one output fires, both inhibitors are excited. This leads to high levels of inhibition, causing firing outputs to stop firing and 'drop out' of the WTA competition. When exactly one output fires, just one of the inhibitors, known as the *stability inhibitor*, is excited. This inhibitor is responsible for maintaining a WTA steady-state: once a single output fires at a time step it becomes the winner of the network. It has a positive feedback self-loop that allows it to keep firing at subsequent times, while all other outputs do not fire due to inhibition from the stability inhibitor.

The basic network construction described above employs biologically plausible structures. In particular, convergence is driven using reciprocal excitatory-inhibitory connections, and stability is maintained via excitatory self-loops. Both these structures are used in many biological models of WTA computation [YG89, CGL92, RB15]. It is widely believed that inhibition is crucial for solving WTA – outputs compete for activation via *lateral inhibition* or *recurrent inhibition* [CGL92, RB15]. In our

[1] Aside from immediate bounds for deterministic circuits using many ($\Omega(n)$) auxiliary neurons [LRMM88, Maa00].

network, this inhibition is achieved through the two auxiliary inhibitors. Previous work has conjectured that widespread use of simple WTA implementations in the brain may explain how complex computation is possible even when inhibition is relatively limited and localized [Maa00]. Our work shows that WTA can be achieved and maintained efficiently using very few inhibitors and with a very simple connectivity structure.

We also demonstrate that, with a larger number of auxiliary neurons, it is possible to obtain faster convergence. In particular, in Section 5.5, we describe, for any input size n and failure probability $\delta > 0$, a family of networks using $O(\log n)$ auxiliary inhibitory neurons which solve the WTA problem in $O(1)$ steps in expectation, and with probability $\geq 1 - \delta$ in $O(\log(1/\delta))$ steps (see Theorems 5.5.2 and 5.5.3). At a high level, more auxiliary inhibitors allow for more fine-tuned levels of inhibition which drive faster convergence. In Section 5.5.7 we sketch two constructions that allow for more general runtime-inhibitor tradeoffs, interpolating between our two-inhibitor and $O(\log n)$-inhibitor constructions.

Lower Bounds

Aside from the above network constructions and runtime analysis, we also prove lower bounds, showing that these constructions are optimal or near optimal. In Section 5.4 we prove that no network can solve WTA (in a reasonable parameter regime) using just a single auxiliary neuron (see Theorem 5.4.3). Roughly, it is not possible for a single neuron to both drive fast convergence and maintain stability of a valid WTA configuration once one has been reached. The dual role that inhibition plays in two-inhibitor construction of driving convergence and maintaining stability requires at least two inhibitors.

We also show that, considering a slightly restricted class of networks, our two-inhibitor construction is near-optimal. No network with just two-auxiliary neurons can solve WTA with constant probability in $o\left(\frac{\log n}{\log \log n}\right)$ steps (see Theorem 5.4.14). This matches the runtime of our network up to a $O(\log \log n)$ factor.

5.1.4 Road Map

In Section 5.2 we describe our spiking neural network model and specify the WTA problem. In Section 5.3 we describe and analyze the convergence of our simple family of two-inhibitor WTA networks. In Section 5.4 we prove lower bounds that show the near optimality of our two-inhibitor construction. In Section 5.5 we show how to obtain faster convergence using a network construction with $O(\log n)$ auxiliary inhibitors. This construction also requires generalizing our model to allow for a *history period*, over which the firing of a neuron's neighbors can affect its membrane potential. Finally, in Section 5.6 we conclude by discussing open questions arising from our work and possible directions for future work.

5.2 Spiking Neural Network Model

In this section we describe our basic neural network model, which consists of a set of neurons connected by weighted synaptic connections. Each neuron fires (spikes) stochastically at each time step, with probability dependent on the firing of its neighbors in the previous time step. These neighbors may have either an excitatory (inducing more firing) or inhibitory (suppressing firing) effect. In Section 5.5 we describe a variation on this model in which the probability that a neuron spikes depends not just on the spiking of its neighbors in the previous time step, but on the spikes during some history period preceding the current time.

5.2.1 Network Structure

We first describe the basic network structure and parameters. A *Spiking Neural Network* (SNN) $\mathcal{N} = \langle N, w, b, f\rangle$ consists of:

- N, a set of neurons, partitioned into a set of input neurons X, a set of output neurons Y, and a set of auxiliary neurons A. N is also partitioned into a set of *excitatory* and *inhibitory* neurons E and I. All input and output neurons are excitatory. That is, $X \cup Y \subseteq E$.

- $w : N \times N \to \mathbb{R}$, a weight function describing the weighted synaptic connections between the neurons in the network. w is restricted in a few notable ways:

  - $w(u, x) = 0$ for all $u \in N$, $x \in X$.
  - Each excitatory neuron $v \in E$ has $w(v, u) \geq 0$ for every u. Each inhibitory neuron $v \in I$ has $w(v, u) \leq 0$ for every u.

- $b : N \to \mathbb{R}$, a bias function, assigning an activation bias to each neuron.

- $f : \mathbb{R} \to [0, 1]$, a spike probability function, satisfying a few restrictions:

  - f is continuous and monotonically increasing.
  - $\lim_{x\to\infty} f(x) = 1$ and $\lim_{x\to-\infty} f(x) = 0$.

Remarks on Network Structure

Before describing the dynamics of our neural network, we give a few remarks on, and explanations of, the above parameters determining the network structure.

Weight Function (w): The weight function w describes the strength of the synaptic connections between neurons in N. The restriction that $w(u, x) = 0$ for every input neuron $x \in X$ is motivated by the desire for networks to be composable. The input neurons in X may be output neurons of another network, and so incoming connections are avoided to simplify definitions and analysis when networks are composed in higher level modular designs.

The restriction that each neuron v is either inhibitory or excitatory is motivated by the observation, known as *Dale's principle*, that neurons typically employ the same neurotransmitter at each outgoing synapse, regardless of its target [Osb13]. Thus, all outgoing connections are either inhibitory or excitatory, depending on the transmitter used. For example, inhibitory connections predominantly stem from inhibitory *GABAergic neurons*, which employ the neurotransmitter *gamma*-Aminobutyric acid (GABA) [WMK+02, RFLHL11].

We often view the weight function as defining the edge weights of a directed graph, whose edges are the synaptic connections. Formally we can define:

Definition 5.2.1 (Synaptic Connection Graph)**.** *Given spiking neural network $\mathcal{N} = \langle N, w, b, f\rangle$, let $G(\mathcal{N})$ be the weighted directed graph with vertex set N and a directed edge (u, v) with weight $w(u, v)$ for all u, v with $w(u, v) \neq 0$.*

Note that the weight function $w(u, v)$ need not be symmetric, and typically will not be. Additionally, we allow $u \in N$ with $w(u, u) \neq 0$. That is, $G(\mathcal{N})$ may have self-loops.

Bias Function (b): The bias function, along with the spike probability function, determines how large a neuron's membrane potential must be for the neuron to spike with good probability. The larger the bias, the more excited the neuron must be before it fires. We will see in Section 5.2.2 exactly how the bias affects the spiking probability.

Spike Probability Function (f): Common choices for the spike probability function f are symmetric functions with $f(0) = \frac{1}{2}$. For example, we will typically set f to the sigmoid function $f(x) = \frac{1}{1+e^{-x/\lambda}}$ for some *temperature parameter* $\lambda > 0$.

5.2.2 Network Dynamics

We now describe in detail the dynamics of our neural network model.

A *configuration* $C : N \rightarrow \{0, 1\}$ of an SNN $\mathcal{N} = \langle N, w, b, f \rangle$ is a mapping from each neuron in the network to a firing state. $C(u) = 1$ indicates that u fires (i.e., generates a spike). $C(u) = 0$ indicates that it does not fire. We similarly define an *input configuration* $C_X : X \rightarrow \{0, 1\}$ to be a mapping from each input neuron to a firing state and an *output configuration* $C_Y : Y \rightarrow \{0, 1\}$ to be a mapping from each output neuron to a firing state. For any configuration C and set of neurons $M \subseteq N$, we let $C(M)$ be the restriction of C to the domain M.

An SNN evolves in a sequence of discrete, synchronous times, which we label with integers $t = 0, 1, ...$ We denote the configuration at time t by N^t. Similarly, we denote the input and output configurations at time t by $X^t \overset{\text{def}}{=} N^t(X)$ and $Y^t \overset{\text{def}}{=} N^t(Y)$ respectively.

Formally, an *execution* is a finite or infinite sequence of configurations. The *length* of a finite execution $N^0 N^1 ... N^t$ is defined to be $t + 1$. The length of an infinite execution $N^0 N^1 ...$ is defined to be ∞. We analogously define an *input execution* and an *output execution* as a sequence of input configurations $X^0 X^1 ...$ and output configurations $Y^0 Y^1 ...$ respectively.

For each neuron $u \in N$, we use the notation $u^t \overset{\text{def}}{=} N^t(u)$ to denote the firing state of neuron u in the configuration N^t. More generally, for any ordered set of neurons $M = \{m_1, ..., m_n\}$ we let $M^t \in \{0, 1\}^n$ denote the binary vector with m_j^t as its j^{th} entry. For any configuration C, we let $\|C\|_1 = |\{u \in N : C(u) = 1\}|$ denote the number of firing neurons in C.

We will typically use α to denote an execution, and α_X, α_Y to denote an input or output execution respectively. We will use a superscript to denote the length of a finite execution. For any execution α let output(α) be the output execution of the same length obtained by restricting each configuration in α to the output neurons Y.

The behavior of an SNN is determined as follows:

- **Input Neurons**: For each problem we consider, we will specify how the infinite input execution $X^0 X^1 ....$ is determined. In this work, we will typically fix the input so that for each $u \in X$, u^t is constant for all $t \geq 0$. However, we may also specify a distribution from which $X^0 X^1 ....$ is drawn. For example, this sequence may be generated by setting $u^t = 1$ with some probability p_u and $u^t = 0$ with probability $1 - p_u$, independently at random for each $u \in X$ and each time t.

- **Initial Firing States**: For each non-input $u \in N \setminus X$, the firing state u^0 is arbitrary. In this work, we typically show convergence results that hold for all possible settings of these initial states, giving our networks a self-stabilizing property, since they will converge from any arbitrary perturbation of the state (see e.g., Theorem 5.2.8).

- **Firing Dynamics**: For each non-input neuron $u \in N \setminus X$ and every time $t \geq 1$, let $\text{pot}(u,t)$ denote the membrane potential at time t and $p(u,t)$ denote the corresponding firing probability. These values are calculated as:

$$\text{pot}(u,t) = \left(\sum_{v \in N} w(v,u) \cdot v^{t-1} \right) - b(u) \text{ and } p(u,t) = f(\text{pot}(u,t)) \tag{5.1}$$

  where f is the spike probability function. At time t, each non-input neuron u fires independently with probability $p(u,t)$.

Any SSN $\mathcal{N} = \langle N, w, b, f \rangle$, initial configuration N^0, and infinite input execution α_X define a probability distribution over infinite executions, $\mathcal{D}(\mathcal{N}, N^0, \alpha_X)$. This distribution is the natural distribution that follows from applying the stochastic firing dynamics of (5.1). Formally, for any finite execution α, we define the *cone* of executions extending α, $A(\alpha)$, to be the set of all infinite executions that start with α. $\mathcal{D}(\mathcal{N}, N^0, \alpha_X) : \mathcal{F} \to [0,1]$ is a probability measure where the σ-algebra $\mathcal{F}$ consists of all such cones, closed under complement, countable unions, and countable intersections.

Given $\mathcal{D}(\mathcal{N}, N^0, \alpha_X)$ we can also define a distribution $\mathcal{D}_Y(\mathcal{N}, N^0, \alpha_X)$ on infinite output executions. Given any finite output execution α_Y, we define the cone $A(\alpha_Y)$ to be the set of all infinite output executions extending α_Y. We define the σ-algebra $\mathcal{F}_Y$ to be the set of all such cones, closed under complement, countable union, and countable intersection. Finally, for $F_Y \in \mathcal{F}_Y$, we define $\mathcal{D}_Y(\mathcal{N}, N^0, \alpha_X) : \mathcal{F}_Y \to [0,1]$ by:

$$\mathcal{D}_Y(\mathcal{N}, N^0, \alpha_X)[F_Y] = \mathcal{D}(\mathcal{N}, N^0, \alpha_X)[\{\alpha : \text{output}(\alpha) \in F_Y\}].$$

5.2.3 Problems and Solving Problems

A problem P is a mapping from an infinite input execution α_X to a set of output distributions. A network $\mathcal{N}$ is said to *solve problem P on input α_X* if, for any initial configuration N^0, the output distribution $\mathcal{D}_Y(\mathcal{N}, N^0, \alpha_X)$ is an element of $P(\alpha_X)$. A network $\mathcal{N}$ is said to *solve problem P* if it solves P on every infinite input execution α_X. For an example of such a problem definition see Section 5.2.6, where we formally define the winner-take-all problem.

5.2.4 Basic Results and Properties of the Model

In this section we prove some basic properties of the spiking neural network model described in the preceding sections. The first property is a simple Markov independence claim: conditioned on the configuration at time $t-1$, a network's execution from time t on is independent of all times before $t-1$. Formally:

Lemma 5.2.2 (Markov Property)**.** *Let $\mathcal{N} = \langle N, w, b, f \rangle$ be an SNN. For any time $t \geq 1$, and finite execution $C^0 ... C^{t-1}$ of $\mathcal{N}$, and any configuration C of $\mathcal{N}$:*

$$\mathbb{P}[N^t = C | N^{t-1} N^{t-2} ... N^0 = C^{t-1} C^{t-2} ... C^0] = \mathbb{P}[N^t = C | N^{t-1} = C^{t-1}].$$

Proof. The potential of every $u \in N$ at time t as computed in (5.1) is determined by N^{t-1}. Thus, the spike probability $p(u,t) = f(\mathrm{pot}(u,t))$ is fully determined by N^{t-1}.

So, conditioned on $N^{t-1} = C^{t-1}$, $p(u,t)$ is a deterministic function of C^{t-1}. We can compute:

$$\mathbb{P}[N^t = C | N^{t-1} = C^{t-1}] = \prod_{u \in N} \left[C(u) \cdot p(u,t) + (1 - C(u)) \cdot (1 - p(u,t)) \right].$$

So $\mathbb{P}[N^t = C | N^{t-1} = C^{t-1}]$ is a deterministic function of C and all the $p(u,t)$ collectively and thus of C and C^{t-1}. So for *any* $C^0 ... C^{t-2}$,

$$\mathbb{P}[N^t = C | N^{t-1} = C^{t-1}] = \mathbb{P}[N^t = C | N^{t-1} N^{t-2} ... N^0 = C^{t-1} C^{t-2} ... C^0],$$

giving the lemma. □

In our proofs, we will often bound the probability of some event $\mathcal{E}_t$ occurring at time t, giving a bound independent of the preceding network configuration N^{t-1}. However, $\mathcal{E}_t$ itself will depend on N^{t-1}, and there may be correlations between $\mathcal{E}_t$ and $\mathcal{E}_{t'}$ for $t \neq t'$. Below, we give a useful lemma which allows us to bound the number of times that $\mathcal{E}_t$ occurs over a given time period by comparing to the number of times that a coin tossed independently at each time would come up heads in the same time period.

Lemma 5.2.3. *For every $t \in \mathbb{Z}^{>0}$ let $A_t \in \mathcal{A}$ be a random variable in some domain $\mathcal{A}$, $f : \mathcal{A} \to \{0,1\}$ be any function, and $B_t = f(A_t)$. Let $Z_t \in \{0,1\}$ be a set of independent random variables. Suppose:*

1. $\mathbb{P}[B_1 = 1] \geq \mathbb{P}[Z_1 = 1]$.

2. For every $t \geq 2$, $\mathbb{P}[B_t = 1 | A_{t-1}, ..., A_1] \geq \mathbb{P}[Z_t = 1]$.

Then for every t *and* $d \in \mathbb{Z}^{\geq 0}$,

$$\mathbb{P}\left[\sum_{i=1}^{t} B_i \geq d\right] \geq \mathbb{P}\left[\sum_{i=1}^{t} Z_i \geq d\right]. \tag{5.2}$$

Lemma 5.2.3 and its proof are similar to Lemma 2.2 of [KKKL11]. However, we include a full proof for completeness and since we are in a slightly different setting, where we condition on the full past state, rather than just the preceding values of $B_t = f(X_t)$.

Proof. We prove the result via induction on t. The base case with $t = 1$ is given by assumption (1). For any $t > 1$, assuming that (5.2) holds for all $t' < t$, we have:

$$\begin{aligned}
\mathbb{P}\left[\sum_{i=1}^{t} B_i \geq d\right] &= \mathbb{P}\left[B_t = 1 \mid \sum_{i=1}^{t-1} B_i = (d-1)\right] \cdot \mathbb{P}\left[\sum_{i=1}^{t-1} B_i = (d-1)\right] + \mathbb{P}\left[\sum_{i=1}^{t-1} B_i \geq d\right]\\
&\geq \mathbb{P}\left[Z_t = 1\right] \cdot \mathbb{P}\left[\sum_{i=1}^{t-1} B_i = d-1\right] + \mathbb{P}\left[\sum_{i=1}^{t-1} B_i \geq d\right]\\
&= \mathbb{P}\left[Z_t = 1\right] \cdot \left(\mathbb{P}\left[\sum_{i=1}^{t-1} B_i \geq d-1\right] - \mathbb{P}\left[\sum_{i=1}^{t-1} B_i \geq d\right]\right) + \mathbb{P}\left[\sum_{i=1}^{t-1} B_i \geq d\right]\\
&= \mathbb{P}\left[Z_t = 1\right] \cdot \mathbb{P}\left[\sum_{i=1}^{t-1} B_i \geq d-1\right] + \mathbb{P}\left[Z_t = 0\right] \cdot \mathbb{P}\left[\sum_{i=1}^{t-1} B_i \geq d\right].
\end{aligned}$$

By the inductive assumption we can then bound:

$$\begin{aligned}
\mathbb{P}\left[\sum_{i=1}^{t} B_i \geq d\right] &\geq \mathbb{P}\left[Z_t = 1\right] \cdot \mathbb{P}\left[\sum_{i=1}^{t-1} Z_i \geq d-1\right] + \mathbb{P}\left[Z_t = 0\right] \cdot \mathbb{P}\left[\sum_{i=1}^{t-1} Z_i \geq d\right]\\
&= \mathbb{P}\left[\sum_{i=1}^{t} Z_i \geq d\right]
\end{aligned}$$

which gives the lemma. □

We next prove a related theorem, but in a more specialized setting. We consider a set of neurons $\{u_1, ..., u_s\}$ for which we can lower bound the probability of each u_i spiking at time $t+1$ given that it spiked at time t (i.e., given that $u_i^t = 1$). We show that, while the behavior of the neurons may be highly correlated, the number of neurons in the set that spike for t consecutive times can be lower bounded by comparing these neurons to a set of independent random variables with comparable spiking probabilities.

Lemma 5.2.4. *Let $\mathcal{N} = \langle N, w, b, f\rangle$ be an SNN, and let $\{u_1, ..., u_s\} \subseteq N$ be any set of neurons in the network. Let $Z_{i,t} \in \{0, 1\}$ be a set of independent random variables. Suppose that:*

1. *The initial configuration N^0 of $\mathcal{N}$ has $N^0(u_i) = 1$ for every $i \in \{1, ..., s\}$.*
2. *For every $i \in \{1, ..., s\}$, any configuration C of $\mathcal{N}$ with $C(u_i) = 1$, and any $t \geq 0$:*
$$\mathbb{P}[u_i^{t+1} = 1 | N^t = C] \geq \mathbb{P}[Z_{i,t+1} = 1].$$

Let $\mathcal{I}_i(t) \in \{0, 1\}$ be an indicator variable for the event that $u_i^1 = ... = u_i^t = 1$. Let $\bar{\mathcal{I}}_i(t) \in \{0, 1\}$ be an indicator variable for the event that $Z_{i,1} = ... = Z_{i,t} = 1$. Then for every t and $d \in \mathbb{Z}^{\geq 0}$,

$$\mathbb{P}\left[\sum_{i=1}^{s} \mathcal{I}_i(t) \geq d\right] \geq \mathbb{P}\left[\sum_{i=1}^{s} \bar{\mathcal{I}}_i(t) \geq d\right]. \tag{5.3}$$

Proof. We prove the lemma via a coupling argument. At a high level, we define a set of auxiliary random variables $\hat{\mathcal{I}}_i(t)$ for $i \in \{1, ..., s\}$. We construct these random variables such that their joint distribution is *identical* to that of the random variables $\mathcal{I}_i(t)$. Additionally, we correlate $\hat{\mathcal{I}}_i(t)$ with the variables $\{Z_{i,t}\}$ in such a way that we always have $\hat{\mathcal{I}}_i(t) \geq \bar{\mathcal{I}}_i(t)$. We thus have:

$$\mathbb{P}\left[\sum_{i=1}^{s} \mathcal{I}_i(t) \geq d\right] = \mathbb{P}\left[\sum_{i=1}^{s} \hat{\mathcal{I}}_i(t) \geq d\right] \geq \mathbb{P}\left[\sum_{i=1}^{s} \bar{\mathcal{I}}_i(t) \geq d\right], \tag{5.4}$$

which gives the lemma.

Definition of Coupled Random Variables $\hat{\mathcal{I}}_i(t)$.

Given $\mathcal{N}$, the distribution on executions of $\mathcal{N}$ with initial configuration N^0 and input configuration α_X, induced by the update rules described in Section 5.2.2 is given by $\mathcal{D}(\mathcal{N}, N^0, \alpha_X)$. We define the distribution $\hat{\mathcal{D}}(\mathcal{N}, N^0, \alpha_X)$, which is identical to $\mathcal{D}(\mathcal{N}, N^0, \alpha_X)$ except coupled to the auxiliary random variables $\{Z_{i,t}\}$ in the following way:

For any $t \geq 0$, execution $\alpha^t = C^0...C^t$, and $i \in \{1, ..., s\}$ with $C^0(u_i) = ... =$

$C^t(u_i) = 1$ let

$$\epsilon_{i,\alpha^t} = \mathop{\mathbb{P}}_{\mathcal{D}(\mathcal{N},N^0,\alpha_X)}[u_i^{t+1} = 1 | N^0 ... N^t = \alpha^t] - \mathbb{P}[Z_{i,t+1} = 1]. \tag{5.5}$$

By assumption (2) in the lemma statement and Lemma 5.2.2 we have $\epsilon_{i,\alpha^t} \geq 0$. Let $E_{i,\alpha^t} \in \{0,1\}$ be a random variable which is independently set to 1 with probability $\frac{\epsilon_{i,\alpha^t}}{1-\mathbb{P}[Z_{i,t+1}=1]}$ and 0 otherwise. The distribution $\hat{\mathcal{D}}(\mathcal{N}, N^0, \alpha_X)$ is given by iteratively drawing a configuration N^{t+1} in the same way as in $\mathcal{D}(\mathcal{N}, N^0, \alpha_X)$, with spiking probabilities given by the potentials induced by N^t. However, if $i \in \{1, ..., s\}$ and $N^0 ... N^t = \alpha^t$ with $N^0(u_i) = ... = N^t(u_i) = 1$, we set

$$u_i^{t+1} = \max(Z_{i,t+1}, E_{i,\alpha^t}). \tag{5.6}$$

Using (5.5) and the definition of E_{i,α^t} we can see that

$$\begin{aligned}
\mathop{\mathbb{P}}_{\hat{\mathcal{D}}(\mathcal{N},N^0,\alpha_X)}[u_i^{t+1} = 1 | N^0 ... N^t = \alpha^t] &= 1 - \mathop{\mathbb{P}}_{\hat{\mathcal{D}}(\mathcal{N},N^0,\alpha_X)}[u_i^{t+1} = 0 | N^0 ... N^t = \alpha^t] \\
&= 1 - (1 - \mathbb{P}[Z_{i,t+1} = 1]) \cdot \big(1 - \mathbb{P}[E_{i,\alpha^t} = 1]\big) \\
&= 1 - (1 - \mathbb{P}[Z_{i,t+1} = 1]) \cdot \left(1 - \frac{\epsilon_{i,\alpha^t}}{1 - \mathbb{P}[Z_{i,t+1} = 1]}\right) \\
&= \mathbb{P}[Z_{i,t+1} = 1] + \epsilon_{i,\alpha^t} \\
&= \mathop{\mathbb{P}}_{\mathcal{D}(\mathcal{N},N^0,\alpha_X)}[u_i^{t+1} | N^0 ... N^t = \alpha^t]. \qquad \text{(By (5.5))}
\end{aligned}$$

Thus, the probability that any neuron spikes at time t conditioned on the network configuration at all times before t is identical in executions drawn from $\hat{\mathcal{D}}(\mathcal{N}, N^0, \alpha_X)$ and $\mathcal{D}(\mathcal{N}, N^0, \alpha_X)$. So for any t, if $N^0 ... N^t$ is drawn from $\mathcal{D}(\mathcal{N}, N^0, \alpha_X)$ and $\hat{N}^0 ... \hat{N}^t$ from $\hat{\mathcal{D}}(\mathcal{N}, N^0, \alpha_X)$, these two executions are identically distributed. In particular, if $\hat{\mathcal{I}}_i(t) \in \{0,1\}$ is an indicator variable for the event that $\hat{N}^1(u_i) = ... = \hat{N}^t(u_i) = 1$, then $\hat{\mathcal{I}}_i(t)$ and $\mathcal{I}_i(t)$ are identically distributed.

Proof that Coupled Random Variables Upper Bound Independent Variables.

We can see that $\hat{\mathcal{I}}_i(t) \geq \bar{\mathcal{I}}_i(t)$ via an inductive argument. In the base case, since we assume $N^0(u_i) = 1$ for all $i \in \{1, ..., s\}$, we apply (5.6) to generate $\hat{N}^1(u_i)$. We set $\hat{N}^1(u_i) = \max(Z_{i,1}, E_{i,\alpha^0}) \geq Z_{i,1}$ which gives $\hat{\mathcal{I}}_i(1) \geq \bar{\mathcal{I}}_i(1)$. For $t \geq 1$, if $\bar{\mathcal{I}}_i(t) = 0$ the claim holds trivially since $\hat{\mathcal{I}}_i(t), \bar{\mathcal{I}}_i(t) \in \{0,1\}$. Otherwise, we have $\bar{\mathcal{I}}_i(t) = 1$ which implies that $\bar{\mathcal{I}}_i(t-1) = 1$ and so $\hat{\mathcal{I}}_i(t-1) = 1$ by the inductive assumption. If $\hat{\mathcal{I}}_i(t-1) = 1$ then again we apply (5.6) to generate $\hat{N}^t(u_i)$ and so have

$\hat{N}^t(u_i) = \max(Z_{i,t}, E_{i,\alpha^{t-1}}) \geq Z_{i,t}$, giving $\hat{\mathcal{I}}_i(t) \geq \bar{\mathcal{I}}_i(t)$.

Since $\hat{\mathcal{I}}_i(t)$ is identically distributed to $\mathcal{I}_i(t)$ this completes the lemma by (5.4). □

Our next lemma pertains specifically to networks with a sigmoid spike probability function, $f(x) = \frac{1}{1+e^{-x/\lambda}}$, which we use throughout this chapter. We show that given a network with temperature parameter $\lambda > 0$, we can construct a network with an identical execution distribution for any $\hat{\lambda} > 0$. Thus, we will always consider the case of $\lambda = 1$, which implies the existence of networks satisfying all bounds given for all $\lambda > 0$.

Lemma 5.2.5 (Equivalence of Temperature Parameters)**.** *For $\lambda, \hat{\lambda} > 0$, let $f(x) = \frac{1}{1+e^{-x/\lambda}}$ and $\hat{f}(x) = \frac{1}{1+e^{-x/\hat{\lambda}}}$. Given $\mathcal{N} = \langle N, w, b, f\rangle$, let $\widehat{\mathcal{N}} = \langle N, \hat{w}, \hat{b}, \hat{f}\rangle$ where for all $u, v \in N$, $\hat{w}(u,v) = w(u,v) \cdot \frac{\lambda}{\hat{\lambda}}$ and $\hat{b}(u) = b(u) \cdot \frac{\lambda}{\hat{\lambda}}$. For any length initial configuration N^0 and any infinite input execution α_X:*

$$\mathcal{D}(\mathcal{N}, N^0, \alpha_X) = \mathcal{D}(\widehat{\mathcal{N}}, N^0, \alpha_X).$$

Proof. For any $t \geq 1$ and any configuration C, we can compute the probability that $\mathcal{N}$ is in this configuration at time t conditioned on all past configurations as:

$$\mathop{\mathbb{P}}_{\mathcal{D}(\mathcal{N},N^0,\alpha_X)}[N^t = C | N^{t-1}...N^0] = \prod_{u \in N} [C(u) \cdot p(u,t) + (1 - C(u)) \cdot (1 - p(u,t))] \tag{5.7}$$

where $p(u,t) = f(\text{pot}(u,t))$. Fixing $N^{t-1}...N^0$, we can see that the potential computation (5.1) is a linear function of $w(u,v)$ and $b(u)$ for all $u, v \in N$. Thus, letting $\widehat{\text{pot}}(u,t)$ be the potential of u at time t in $\widehat{\mathcal{N}}$ given $N^{t-1}...N^0$, since $\hat{w}(u,v) = w(u,v) \cdot \frac{\lambda}{\hat{\lambda}}$ and $\hat{b}(u) = b(u) \cdot \frac{\lambda}{\hat{\lambda}}$,

$$\widehat{\text{pot}}(u,t) = \text{pot}(u,t) \cdot \frac{\lambda}{\hat{\lambda}}.$$

This gives that the probability of u spiking at time t in $\widehat{\mathcal{N}}$ given $N^{t-1}...N^0$ equals :

$$\hat{p}(u,t) = \hat{f}(\widehat{\text{pot}}(u,t)) = \hat{f}\left(\text{pot}(u,t) \cdot \frac{\lambda}{\hat{\lambda}}\right) = f(\text{pot}(u,t)) = p(u,t).$$

And since $\hat{p}(u,t) = p(u,t)$ for all $u \in N$ we have using (5.7), for any C,

$$\mathop{\mathbb{P}}_{\mathcal{D}(\mathcal{N},N^0,\alpha_X)}[N^t = C | N^{t-1}...N^0] = \mathop{\mathbb{P}}_{\mathcal{D}(\widehat{\mathcal{N}},N^0,\alpha_X)}[N^t = C | N^{t-1}...N^0]. \tag{5.8}$$

Finally inducting on t we can show that for any finite execution $C^0...C^t$:

$$\underset{\mathcal{D}(\mathcal{N},N^0,\alpha_X)}{\mathbb{P}}[N^0...N^t = C^0...C^t] = \underset{\mathcal{D}(\widehat{\mathcal{N}},N^0,\alpha_X)}{\mathbb{P}}[N^0...N^t = C^0...C^t]. \tag{5.9}$$

This holds trivially for $t = 0$ since

$$\underset{\mathcal{D}(\mathcal{N},N^0,\alpha_X)}{\mathbb{P}}[N^0 = C^0] = \underset{\mathcal{D}(\widehat{\mathcal{N}},N^0,\alpha_X)}{\mathbb{P}}[N^0 = C^0] = 1$$

if $N^0 = C^0$. Both probabilities are zero otherwise. For $t \geq 1$, assume that (5.9) holds for all $t' < t$. Combined with (5.8) this gives:

$$\begin{aligned}\underset{\mathcal{D}(\mathcal{N},N^0,\alpha_X)}{\mathbb{P}}[N^0...N^t = C^0...C^t] &= \underset{\mathcal{D}(\mathcal{N},N^0,\alpha_X)}{\mathbb{P}}[N^0...N^{t-1} = C^0...C^{t-1}] \\ &\quad \cdot \underset{\mathcal{D}(\mathcal{N},N^0,\alpha_X)}{\mathbb{P}}[N^t = C^t | N^0...N^{t-1} = C^0...C^{t-1}] \\ &= \underset{\mathcal{D}(\widehat{\mathcal{N}},N^0,\alpha_X)}{\mathbb{P}}[N^0...N^{t-1} = C^0...C^{t-1}] \\ &\quad \cdot \underset{\mathcal{D}(\widehat{\mathcal{N}},N^0,\alpha_X)}{\mathbb{P}}[N^t = C^t | N^{t-1}...N^0 = C^0...C^{t-1}] \\ &= \underset{\mathcal{D}(\widehat{\mathcal{N}},N^0,\alpha_X)}{\mathbb{P}}[N^0...N^t = C^0...C^t].\end{aligned}$$

This completes the lemma. □

5.2.5 Potential Modifications to the Basic Model

There are many potential modifications to the basic network model described in Sections 5.2.1 and 5.2.2 which may be interesting to consider in future work. We present some below.

- One interesting extension is to add a history period $h > 1$ to the network, so that the spiking probability of a neuron at time t depends on the configuration of the network at times $t-1, ..., t-h$. In Section 5.5, for example, we use a model with history period $h = 2$ to design very fast WTA networks.

- We could consider a very general history model, defining

  $$\text{pot}(u,t) = f(u, N^{t-1}, ..., N^{t-h})$$

  where f is any function.

- A history period h can be thought of as giving each neuron access to a length h queue of firing patterns on which $\text{pot}(u, t)$ depends. It may be interesting to model such a queue as residing within the neuron's state. We could also consider neurons with other types of state, capturing various types of observed biological phenomena. For example, we could model a refractory period, in which a neuron cannot fire again for a certain number of time steps after firing [And03, Izh04].

- A history period may be used, for example, to model a universal decay in the influence of spikes over time. We may specify a non-increasing weight vector $= (c_1, c_2, ..., c_h) \in \mathbb{R}^{\geq 0}$ and modify the potential computation of (5.1) such that for any $t \geq h$:

$$\text{pot}(u, t) = \left(\sum_{i=1}^{h} \sum_{v \in N} c_i \cdot w(v, u) \cdot v^{t-i} \right) - b(u).$$

- It may be interesting to consider networks with neurons of multiple types, with different firing dynamics. The human brain contains as many as 10,000 distinct neuron types [Stu]. Understanding how important neuron specialization is and for what reasons it arises is a very interesting question.

- Similarly, we note that there is evidence that Dale's principal can be violated and that some neurons do have both inhibitory and excitatory outgoing connections [Osb79]. Modeling such neurons to better understand their role and importance is an interesting direction.

5.2.6 The Winner-Take-All Problem

We now define the main problem that we consider in this chapter, the binary winner-take-all (WTA) problem. In this problem, given n input neurons, the goal is to converge to a configuration in which a single output neuron, corresponding to a firing input, fires. This neuron is referred to as the 'winner' of the computation. We first define a valid WTA output configuration for a given input configuration:

Definition 5.2.6 (Valid WTA Output Configuration)**.** *Consider any network $\mathcal{N}$ with n input neurons, labeled $x_1, ..., x_n$, and n output neurons, labeled $y_1, ..., y_n$. For any input configuration C_X of $\mathcal{N}$, a* valid WTA output configuration *for C_X is any output configuration C_Y with $C_Y(y_i) \leq C_X(x_i)$ for all $i \in \{1, ..., n\}$ and $\|C_Y\|_1 = \min(1, \|C_X\|_1)$.*

Interpreting the above definition, the restriction that $\|C_Y\|_1 = \min(1, \|C_X\|_1)$ requires that if at least one input fires, exactly one output fires. The condition $C_Y(y_i) \leq C_X(x_i)$ for all i requires that this firing output corresponds to a firing input. If no inputs fire (i.e., if $\|C_X\|_1 = 0$), then no outputs should fire. With this definition, we can define the WTA problem (see Section 5.2.3 for a description of how problems are defined in our SNN model):

Definition 5.2.7 (Winner-Take-All Problem)**.** *Given input size $n \in \mathbb{Z}^{>0}$, convergence time $t_c \in \mathbb{Z}^{>0}$, stability time $t_s \in \mathbb{Z}^{>0}$, and failure probability $\delta > 0$, the winner-take-all problem* $\mathrm{WTA}(n, t_c, t_s, \delta)$ *is defined as follows:*

- *If α_X is an input execution with X^t fixed for all t, the output distribution $\mathcal{D}_Y(\mathcal{N}, N^0, \alpha_X)$ can be any distribution on executions of n output neurons satisfying:*
  - *With probability $\geq 1 - \delta$, there exists some $t \leq t_c$ such that the output configuration is fixed at times $t, t+1, ..., t+t_s$ and is a valid WTA output configuration for X^t (Def. 5.2.6).*
- *If α_X is any other input execution, the output distribution is unconstrained.*

Thus, to solve $\mathrm{WTA}(n, t_c, t_s, \delta)$, with probability $\geq 1 - \delta$, the network must converge to a valid output configuration within t_c time steps and maintain this configuration for t_s time steps.

Due to the random firing behavior of our neurons, the network will eventually move to a different configuration with some probability. However, if the network solves $\mathrm{WTA}(n, t_c, t_s, \delta)$, since convergence is required given any initial configuration N^0, we can show that it must be self-stabilizing. That is, once it leaves a valid output configuration, it will converge again with probability $\geq 1 - \delta$ within t_c steps, and maintain the new valid configuration again for t_s steps. Formally:

Theorem 5.2.8 (Self Stabilization of Winner-Take-All Networks)**.** *If $\mathcal{N}$ solves* $\mathrm{WTA}(n, t_c, t_s, \delta)$ *for input execution α_X with X^t fixed for all t, given any finite execution $C^0...C^t$ of $\mathcal{N}$, conditioned on $N^0...N^t = C^0...C^t$, with probability $\geq 1 - \delta$ there is some time $t' \leq t + t_c$ such that the output configuration for $\mathcal{N}$ is fixed at times $t', t'+1, ..., t'+t_s$ and is a valid WTA output configuration for X^t.*

Proof. Consider the distribution on infinite executions $N^{t+1}N^{t+2}...$ conditioned on $N^0...N^t = C^0...C^t$. Since the configuration at time $t' \geq t+1$ depends only on the configuration at time $t'-1$, this distribution is identical to $\mathcal{D}_Y(\mathcal{N}, C^t, \alpha_X)$.

Thus, if $\mathcal{N}$ solves WTA(n, t_c, t_s, δ) on α_X, conditioned on $N^0...N^t = C^0...C^t$, with probability $1-\delta$, there is some time $t' \leq t + t_c$ in which $\mathcal{N}$ reaches a valid WTA output configuration for X^t and remains there for t_s steps, giving the lemma.

□

We can also define an expected-time version of the winner-take-all problem as follows:

Definition 5.2.9 (Expected-Time Winner-Take-All Problem)**.** *For any infinite input execution $\alpha_X = X^0X^1...$, stability time $t_s \in \mathbb{Z}^{>0}$, and infinite output execution $\alpha_Y = Y^0Y^1...$ define:*

$$t(\alpha_X, t_s, \alpha_Y) = \min\left\{t : Y^t \textit{ is a valid WTA output configuration and } Y^t = ... = Y^{t+t_s}\right\}.$$

Given input size $n \in \mathbb{Z}^{>0}$, convergence time $t_c \in \mathbb{Z}^{>0}$, stability time $t_s \in \mathbb{Z}^{>0}$, the expected-time winner-take-all problem WTA-EXP(n, t_c, t_s) *is defined as follows:*

- *If α_X is an input execution with X^t fixed for all t, the output distribution $\mathcal{D}_Y(\mathcal{N}, N^0, \alpha_X)$ can be any distribution on executions of n output neurons satisfying:*
$$\mathop{\mathbb{E}}_{\mathcal{D}_Y(\mathcal{N},N^0,\alpha_X)} t(\alpha_X, t_s, \alpha_Y) \leq t_c.$$
- *If α_X is any other input execution, the output distribution is unconstrained.*

5.3 A Two-Inhibitor Solution to the WTA Problem

We now present a simple solution to the WTA problems in Definitions 5.2.7 and 5.2.9 in networks with spike probability given by a sigmoid function. We begin by defining a family of networks $\mathcal{T}_{n,\gamma}$ for any input size n and weight scaling parameter $\gamma \in \mathbb{R}^+$ that solve these problems.

5.3.1 Network Definition

We first give a full definition of our family of two-inhibitor WTA networks, before describing the intuition behind why these networks solve the WTA problem (Definitions 5.2.7 and 5.2.9).

Definition 5.3.1 (Two-Inhibitor WTA Network)**.** *For any positive integer n and $\gamma \in \mathbb{R}^+$, let $\mathcal{T}_{n,\gamma} = \langle N, w, b, f\rangle$ where the spike probability, weight, and bias functions are defined as follows:*

- *The spike probability function f is defined to be the basic sigmoid function:*

$$f(x) \overset{\text{def}}{=} \frac{1}{1+e^{-x}}. \tag{5.10}$$

- *The set of neurons N consists of a set of n input neurons X, labeled $x_1, ..., x_n$, a set of n corresponding outputs Y, labeled $y_1, ..., y_n$, and two auxiliary inhibitor neurons labeled a_s, a_c.*

- *The weight function w is given by:*
  - $w(x_i, y_i) = 3\gamma$, *for all i.*
  - $w(y_i, y_i) = 2\gamma$, *for all i.*
  - $w(a_s, y_i) = w(a_c, y_i) = -\gamma$, *for all i.*
  - $w(y_i, a_s) = w(y_i, a_c) = \gamma$, *for all i.*
  - $w(u, v) = 0$ *for any u, v whose connection is not specified above.*

- *The bias function b is given by:*
  - $b(y_i) = 3\gamma$ *for all i.*
  - $b(a_s) = \gamma/2$.
  - $b(a_c) = 3\gamma/2$.

A diagram of $\mathcal{T}_{n,\gamma}$ is shown in Figure 5-1. Note that the two inhibitors a_s and a_c have identical outgoing connections, and differ just in their bias.

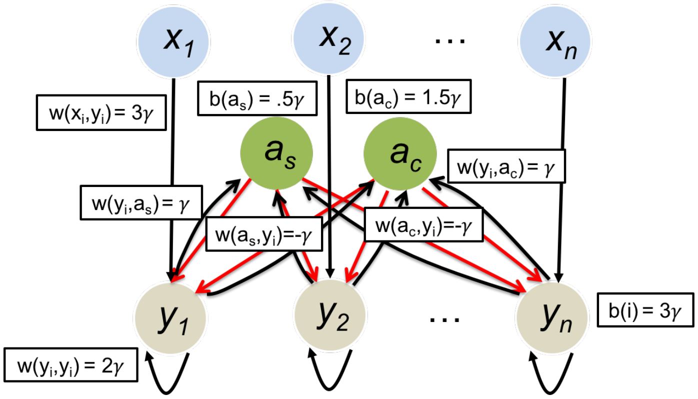

Figure 5-1: Our two-inhibitor WTA network $\mathcal{T}_{n,\gamma}$ as described in Definition 5.3.1.

Intuition Behind the Two-Inhibitor Network

Before giving a formal analysis of the behavior of $\mathcal{T}_{n,\gamma}$, we give some intuition behind why this family of two-inhibitor networks solves the WTA problem. In the description below, we informally refer to events that occur 'with high probability'. We will quantify the meaning of such statements in our full analysis.

In $\mathcal{T}_{n,\gamma}$, each input is connected to its corresponding output with a positive weight. Thus, firing inputs will initially cause their corresponding outputs to fire with high probability. For the network to solve WTA, it must converge to a state in which just a single one of these outputs fires.

To ensure this convergence, $\mathcal{T}_{n,\gamma}$ has two inhibitors $A = \{a_s, a_c\}$. The neuron a_s is a *stability* inhibitor that maintains a valid WTA output configuration once it has been reached. It fires with high probability at time t whenever at least one output fires at time $t-1$. The neuron a_c is a *convergence* inhibitor that fires with high probability whenever at least two outputs fire at time $t-1$.

The weights connecting a_s and a_c to the outputs are set such that when both fire at time t, any output that fired at time t will fire with probability $1/2$ at time $t+1$. Any output that did not fire at time t will not fire at time $t+1$ with high probability. This distinguished behavior between previously firing and non-firing outputs is due to the self-loops on each output neuron, which allow firing outputs to partially overcome the strong inhibition from a_s and a_c.

In this way, if two or more outputs fire at time t, both inhibitors fire with high probability and the high level of inhibition causes outputs to 'drop out of contention' for the winning position with probability $1/2$. After $O(\log n)$ time steps, nearly all the outputs stop firing and, with constant probability, there is a time step in which exactly one output fires. Once this step occurs, with high probability, a_c ceases firing and just a_s fires. This decreased level of inhibition allows the winner to keep firing with high probability, as the inhibition is fully offset by the winner's excitatory self-loop. However, with high probability, the inhibition prevents any other output whose excitatory self-loop is inactive from firing. Thus the network remains in the valid WTA output configuration for a large number of time steps with high probability.

In the event that a time in which a single output fires does not occur, then the network 'resets'. No outputs fire at some time, causing the inhibitors to both cease firing. Thus, all outputs with firing inputs are able to fire, and convergence starts again. Within $O(\log 1/\delta)$ of these resets each reaching a valid WTA output state with constant probability, the network reaches a valid WTA output state with probability $\geq 1-\delta$ and so solves the WTA problem of Definition 5.2.7. Similarly, the network requires $O(1)$ resets in expectation to reach a valid WTA output state, giving a solution to the expected-time version of the problem in Definition 5.2.9. Formally, we will prove the following:

Theorem 5.3.2 (Two-Inhibitor WTA). *For $\gamma \geq 4\ln((n+2)t_s/\delta)+10$, $\mathcal{T}_{n,\gamma}$ solves* $\text{WTA}(n, t_c, t_s, \delta)$ *for any $t_c \geq 72(\log_2 n + 1) \cdot (\log_2(1/\delta)+1)$.*

Theorem 5.3.3 (Two-Inhibitor Expected-Time WTA). *For $\gamma \geq 4\ln((n+2)t_s)+10$, $\mathcal{T}_{n,\gamma}$ solves* $\text{WTA-EXP}(n, t_c, t_s)$ *for any $t_c \geq 108(\log_2 n + 3)$.*

Proof Roadmap. We prove Theorems 5.3.2 and 5.3.3 in in Sections 5.3.2-5.3.7. The analysis is broken down as follows:

Section 5.3.2: Prove basic *one-step lemmas* which characterize single time step transitions of $\mathcal{T}_{n,\gamma}$, showing that the neurons behave as described in the above high-level description.

Section 5.3.3: Prove that, once in a valid WTA configuration, $\mathcal{T}_{n,\gamma}$ stays in this configuration with high probability (that is, valid WTA configurations are stable).

Section 5.3.4: Show that all configurations of $\mathcal{T}_{n,\gamma}$ transition with high probability within two time steps to a small set of *good configurations*, from which we will prove fast convergence.

Section 5.3.5 Show basic transition lemmas for this set of good configurations, characterizing the network's behavior at the times immediately following a good configuration.

Section 5.3.6 Use the above transition lemmas to show that the network converges, with constant probability, from any good configuration (and hence any configuration by Section 5.3.4) to a valid WTA configuration within $O(\log n)$ time steps.

Section 5.5.6 Complete the analysis, demonstrating with what parameters $\mathcal{T}_{n,\gamma}$ solves the winner-take-all problem (Definitions 5.2.7 and 5.2.9).

5.3.2 Basic Results and One-step Lemmas

We begin with some basic results that will be important throughout our analysis, including a few 'one-step' lemmas, which characterize the transition probabilities from a set of configurations at time t to a set of configurations at time $t+1$.

We first show that, unless a neuron has potential 0, either it fires with high probability (i.e., except with probability that is inverse exponential in γ) or it does not fire with high probability.

Lemma 5.3.4 (Characterization of Firing Probabilities)**.** *For any time $t \geq 1$ and any $u \in N$:*

$$\begin{aligned} &\textit{If } \operatorname{pot}(u,t) = 0, \textit{ then } p(u,t) = 1/2. \\ &\textit{If } \operatorname{pot}(u,t) < 0, \textit{ then } p(u,t) \leq e^{-\gamma/2}. \\ &\textit{If } \operatorname{pot}(u,t) > 0, \textit{ then } p(u,t) \geq 1 - e^{-\gamma/2}. \end{aligned}$$

Proof. If $\operatorname{pot}(u,t) = 0$, then by (5.1) $p(u,t) = f(\operatorname{pot}(u,t)) = \frac{1}{1+e^0} = 1/2$. Otherwise consider the potential calculation of (5.1) in the case when $h = 1$:

$$\operatorname{pot}(u,t) = \sum_{v \in N} w(v,u) \cdot v^{t-1} - b(u).$$

By Definition 5.3.1, for all u, v, $w(v,u)$ and $b(u)$ are integer multiples of $\gamma/2$. Thus, since $v^{t-1} \in \{0,1\}$, $\operatorname{pot}(u,t)$ is also an integer multiple of $\gamma/2$. So, if $\operatorname{pot}(u,t) < 0$, then $\operatorname{pot}(u,t) \leq -\gamma/2$ and:

$$p(u,t) = f(\operatorname{pot}(u,t)) \leq f(-\gamma/2) = \frac{1}{1+e^{\gamma/2}} \leq e^{-\gamma/2}.$$

Similarly, if $\operatorname{pot}(u,t) > 0$, then $\operatorname{pot}(u,t) \geq \gamma/2$ and so:

$$p(u,t+1) = f(\operatorname{pot}(u,t)) \geq f(\gamma/2) = \frac{1}{1+e^{-\gamma/2}} \geq 1 - e^{-\gamma/2}.$$

□

We next show that if output y_i does not correspond to a firing input (i.e., $x_i^t = 0$), then starting from any configuration of $\mathcal{T}_{n,\gamma}$ at time t, with high probability y_i does not fire at time $t+1$. That is, with high probability, outputs that are not valid winners of the WTA computation do not fire.

Lemma 5.3.5 (Correct Output Behavior)**.** *For any time t, any configuration C of $\mathcal{T}_{n,\gamma}$, and any i with $C(x_i) = 0$,*

$$\mathbb{P}[y_i^{t+1} = 1 | N^t = C] \leq e^{-\gamma/2}.$$

Proof. If $N^t = C$ then $x_i^t = C(x_i)$. We can compute y_i's potential at time $t+1$, assuming $x_i^t = 0$:

$$\begin{aligned}\text{pot}(y_i, t+1) &= w(x_i, y_i)x_i^t + w(y_i, y_i)y_i^t + w(a_s, y_i)a_s^t + w(a_c, y_i)a_c^t - b(y_i)\\ &\leq 0 + 2\gamma + 0 + 0 - 3\gamma = -\gamma.\end{aligned}$$

Thus, by Lemma 5.3.4, since $\text{pot}(y_i, t+1) < 0$, $p(y_i, t+1) \leq e^{-\gamma/2}$. □

Applying Lemma 5.3.5 and a simple union bound over all n outputs yields the following corollary:

Corollary 5.3.6 (Correct Output Behavior, All Neurons)**.** *For any time t and configuration C of $\mathcal{T}_{n,\gamma}$,*

$$\mathbb{P}[y_i^{t+1} \leq x_i^t \textit{ for all } i | N^t = C] \geq 1 - ne^{-\gamma/2}.$$

Proof. If $C(x_i) = 1$ then conditioned on $N^t = C$, $x_i^t = 1$ and so $y_i^{t+1} \leq x_i^t$ always. Otherwise, by Lemma 5.3.5, if $C(x_i) = 0$, then $\mathbb{P}[y_i^{t+1} = 0 | N^t = C] \geq 1 - e^{-\gamma/2}$. Union bounding over all such inputs (of which there are at most n) gives the corollary. □

We next show that the inhibitors a_s and a_c behave as expected with high probability.

Lemma 5.3.7 (Correct Inhibitor Behavior)**.** *For any time t and configuration C of $\mathcal{T}_{n,\gamma}$,*

1. *If $\|C(Y)\|_1 = 0$, then $\mathbb{P}[a_s^{t+1} = a_c^{t+1} = 0 | N^t = C] \geq 1 - 2e^{-\gamma/2}$.*
2. *If $\|C(Y)\|_1 = 1$, then $\mathbb{P}[a_s^{t+1} = 1$ and $a_c^{t+1} = 0 | N^t = C] \geq 1 - 2e^{-\gamma/2}$.*
3. *If $\|C(Y)\|_1 \geq 2$, then $\mathbb{P}[a_s^{t+1} = 1 = a_c^{t+1} = 1 | N^t = C] \geq 1 - 2e^{-\gamma/2}$.*

Proof. We prove each case above separately. Note that, conditioned on $N^t = C$, $Y^t = C(Y)$.

Case 1: $\|C(Y)\|_1 = 0$**.**

In this case, the inhibitors receive no excitatory signal from the outputs so,

$$\text{pot}(a_s, t+1) = -b(a_s) < 0 \quad \text{and} \quad \text{pot}(a_c, t+1) = -b(a_c) < 0.$$

Thus by Lemma 5.3.4 and a union bound over the two inhibitors,

$$\mathbb{P}[a_s^{t+1} = a_c^{t+1} = 0 | N^t = C] \geq 1 - 2e^{-\gamma/2}.$$

Case 2: $\|C(Y)\|_1 = 1$**.**

In this case we have:

$$\begin{aligned}
\text{pot}(a_s, t+1) &= \sum_{j=1}^{n} w(y_j, a_s) y_j^t - b(a_s) \\
&= \gamma - \gamma/2 = \gamma/2. \\
\text{pot}(a_c, t+1) &= \sum_{j=1}^{n} w(y_j, a_c) y_j^t - b(a_c) \\
&= \gamma - 3\gamma/2 = -\gamma/2.
\end{aligned}$$

Again by Lemma 5.3.4 and a union bound, $\mathbb{P}[a_s^{t+1} = 1 \text{ and } a_c^{t+1} = 0 | N^t = C] \geq 1 - 2e^{-\gamma/2}$.

Case 3: $\|C(Y)\|_1 \geq 2$

Finally, in this case:

$$\begin{aligned}
\text{pot}(a_s, t+1) &= \sum_{j=1}^{n} w(y_j, a_s) y_j^t - b(a_s) \\
&\geq 2\gamma - \gamma/2 = 3\gamma/2. \\
\text{pot}(a_c, t+1) &= \sum_{j=1}^{n} w(y_j, a_c) y_j^t - b(a_c) \\
&\geq 2\gamma - 3\gamma/2 = \gamma/2.
\end{aligned}$$

So by Lemma 5.3.4 and a union bound, $\mathbb{P}[a_s^{t+1} = a_c^{t+1} = 1 | N^t = C] \geq 1 - 2e^{-\gamma/2}$, completing the lemma. □

Combined with Corollary 5.3.6, Lemma 5.3.7 conclusion (1) gives:

Lemma 5.3.8 (Quiescent Behavior)**.** *Assume the input execution α_X of $\mathcal{T}_{n,\gamma}$ has X^t fixed for all t and $\|X^t\|_1 = 0$. For any time t and configuration C with $C(X) = X^t$,*

$$\mathbb{P}[\|N^{t+2}\|_1 = 0 | N^t = C] \geq 1 - 2(n+1)e^{-\gamma/2}.$$

Proof. Let $\mathcal{E}_{10}$ be the event that $N^t = C$ and $\|Y^{t+1}\|_1 = 0$. Let $\mathcal{E}_{20}$ be the event that $\|N^{t+2}\|_1 = 0$. That is, that no neurons fire at time $t+2$. Conditioned on $\mathcal{E}_{10}$, by Lemma 5.3.7 conclusion (1), with probability $\geq 1 - 2e^{-\gamma/2}$, $a_s^{t+2} = a_c^{t+2} = 0$. Again by Corollary 5.3.6, conditioned on $\mathcal{E}_{10}$, with probability $\geq 1 - ne^{-\gamma/2}$, $\|Y^{t+2}\|_1 = 0$. So, overall by a union bound,

$$\mathbb{P}[\mathcal{E}_{20}|\mathcal{E}_{10}] \geq 1 - (n+2)e^{-\gamma/2}.$$

By Corollary 5.3.6, since $\|X^t\|_1 = 0$, $\mathbb{P}[\mathcal{E}_{10}|N^t = C] \geq 1 - ne^{-\gamma/2}$. We can thus bound:

$$\begin{aligned}
\mathbb{P}[\mathcal{E}_{20}|N^t = C] &\geq \mathbb{P}[\mathcal{E}_{10}|N^t = C] \cdot \mathbb{P}[\mathcal{E}_{20}|\mathcal{E}_{10}, N^t = C] \\
&= \mathbb{P}[\mathcal{E}_{10}|N^t = C] \cdot \mathbb{P}[\mathcal{E}_{20}|\mathcal{E}_{10}] \\
&\qquad\qquad \text{(Since, by definition, } \mathcal{E}_{10} \text{ implies } N^t = C.) \\
&\geq \left(1 - ne^{-\gamma/2}\right) \cdot \left(1 - (n+2)e^{-\gamma/2}\right) \\
&\geq 1 - 2(n+1)e^{-\gamma/2},
\end{aligned}$$

which gives the lemma. □

We next show that the stability inhibitor, with high probability, induces exactly the outputs that fired at the previous time step to fire in the next step. We show the lemma in fact for any configuration in which exactly one inhibitor fires. Since a_s and a_c have identical outgoing edges, they have a symmetric effect on the firing probabilities of other neurons.

Lemma 5.3.9 (Stability Inhibitor Effect). *For any time t and configuration C of $\mathcal{T}_{n,\gamma}$ with ($C(a_s) = 1$ and $C(a_c) = 0$) or ($C(a_s) = 0$ and $C(a_c) = 1$) and $C(y_i) \leq C(x_i)$ for all i,*

$$\mathbb{P}[Y^{t+1} = Y^t | N^t = C] \geq 1 - ne^{-\gamma/2}.$$

Proof. For any configuration C with $C(a_c) = 1$ and $C(a_s) = 0$, let $\bar{C}$ denote the configuration with $\bar{C}(a_c) = 0$, $\bar{C}(a_s) = 1$, and $\bar{C}(u) = C(u)$ for all other $u \in N \setminus \{a_c, a_s\}$. Since a_c and a_s have no self-loops and have identical outgoing connections, the distribution of N^{t+1} given $N^t = C$ is identical to its distribution given $N^t = \bar{C}$. Thus, we can assume without loss of generality in the proof of this lemma that $C(a_s) = 1$ and $C(a_c) = 0$.

Conditioned on $N^t = C$, $y_i^t \leq x_i^t$ by assumption. So for any output with $y_i^t = 1$, we have $x_i^t = 1$. This gives:

$$\begin{aligned}\text{pot}(y_i, t+1) &= w(x_i, y_i)x_i^t + w(y_i, y_i)y_i^t + w(a_s, y_i)a_s^t + w(a_c, y_i)a_c^t - b(y_i)\\ &= 3\gamma + 2\gamma - \gamma + 0 - 3\gamma\\ &= \gamma.\end{aligned}$$

In contrast, for any output with $y_i^t = 0$:

$$\begin{aligned}\text{pot}(y_i, t+1) &= w(x_i, y_i)x_i^t + w(y_i, y_i)y_i^t + w(a_s, y_i)a_s^t + w(a_c, y_i)a_c^t - b(y_i)\\ &\leq 3\gamma + 0 - \gamma + 0 - 3\gamma\\ &= -\gamma.\end{aligned}$$

Thus, by Lemma 5.3.4, if $y_i^t = 1$, then $y_i^{t+1} = 1$ with probability $\geq 1 - e^{-\gamma/2}$. If $y_i^t = 0$, then $y_i^{t+1} = 0$ with probability $\geq 1 - e^{-\gamma/2}$. The lemma follows after union bounding over all n outputs. □

Finally, we show that when both the stability and convergence inhibitors fire at time t, not only do outputs not firing at time t not fire at time $t+1$ with high probability, but also, all firing outputs at time t stop firing with probability $1/2$ at time $t+1$. Conditioned on the configuration at time t, these outputs fire independently, a property which will be useful in our eventual proof of progress towards a valid WTA configuration in Lemma 5.3.23.

Lemma 5.3.10 (Convergence Inhibitor Effect)**.** *For any time t and configuration C of $\mathcal{T}_{n,\gamma}$ with $C(a_s) = C(a_c) = 1$ and $C(y_i) \leq C(x_i)$ for all i,*

1. *$\mathbb{P}[y_i^{t+1} \leq y_i^t$ for all $i | N^t = C] \geq 1 - ne^{-\gamma/2}$.*
2. *If $y_i^t = 1$, $\mathbb{P}[y_i^{t+1} = 1 | N^t = C] = 1/2$.*
3. *For $i \neq j$, y_i^{t+1} and y_j^{t+1} are independent conditioned on $N^t = C$.*

Proof. Conditioned on $N^t = C$, if $y_i^t = 1$, by assumption $x_i^t = 1$. We can thus compute:

$$\begin{aligned}\text{pot}(y_i, t+1) &= w(x_i, y_i)x_i^t + w(y_i, y_i)y_i^t + w(a_s, y_i)a_s^t + w(a_c, y_i)a_c^t - b(y_i)\\ &= 3\gamma + 2\gamma - \gamma - \gamma - 3\gamma\\ &= 0.\end{aligned}$$

We thus have $\mathbb{P}[y_i^{t+1} = 1 | N^t = C] = 1/2$ by Lemma 5.3.4. This gives conclusion (2). Conclusion (3) holds trivially since, with N^t fixed, u^{t+1} is independent of v^{t+1} for all $u \neq v$.

We can also bound if $y_i^t = 0$:

$$\begin{aligned}
\text{pot}(y_i, t+1) &= w(x_i, y_i)x_i^t + w(y_i, y_i)y_i^t + w(a_s, y_i)a_s^t + w(a_c, y_i)a_c^t - b(y_i) \\
&\leq 3\gamma + 0 - \gamma - \gamma - 3\gamma \\
&= -2\gamma.
\end{aligned}$$

Thus, by Lemma 5.3.4, $\mathbb{P}[y_i^{t+1} = 1 | N^t = C] \leq e^{-\gamma/2}$. By a union bound over at most n such outputs, we have, with probability $\geq 1 - ne^{-\gamma/2}$, $y_i^{t+1} \leq y_i^t$ for all i , completing the lemma. □

5.3.3 Stability

In this section we extend our definition of a valid WTA output configuration (Definition 5.2.6), to give a more restrictive notion of a valid WTA configuration, which additionally requires that the auxiliary neurons a_s, a_c are in a good state. We show that once the network is in such a state at time t, it remains there with high probability at time $t+1$.

Definition 5.3.11 (Valid WTA Configuration). *A valid WTA configuration of $\mathcal{T}_{n,\gamma}$ is a configuration C with $C(y_i) \leq C(x_i)$ for all $i \in \{1, ..., n\}$ and $\|C(Y)\|_1 = \min(1, \|C(X)\|_1)$ (i.e., the outputs satisfy Definition 5.2.6) and further, $C(a_c) = 0$ and $C(a_s) = \min(1, \|C(X)\|_1)$.*

In the above we require $C(a_s) = \min(\|C(X)\|_1, 1)$. That is, the stability inhibitor fires in a valid WTA configuration, unless no inputs fire. If no inputs fire, a valid WTA configuration requires that neither a_s nor a_c fire and additionally, that no outputs fire.

Lemma 5.3.12 (Stability of Valid Configurations). *Assume the input execution α_X of $\mathcal{T}_{n,\gamma}$ has X^t fixed for all t. For any time t and valid WTA configuration C with $C(X) = X^t$,*

$$\mathbb{P}[N^{t+1} = N^t | N^t = C] \geq 1 - (n+2)e^{-\gamma/2}.$$

Proof. By Definition 5.3.11, since C is a valid WTA configuration, we have

$$\|C(Y)\|_1 = \min(1, \|C(X)\|_1) \in \{0, 1\}.$$

We prove the lemma via a case analysis on $\|C(Y)\|_1$.

Case 1 : $\|C(Y)\|_1 = 0$

In this case, since C is a valid WTA configuration, according to Definition 5.3.11, conditioned on $N^t = C$, we must have $\|X^t\|_1 = 0$ and $a_s^t = a_c^t = 0$. By Corollary 5.3.6, since $\|X^t\|_1 = 0$,

$$\mathbb{P}[\|Y^{t+1}\|_1 = 0 | N^t = C] \geq 1 - ne^{-\gamma/2}.$$

By Lemma 5.3.7 conclusion (1), since $\|C(Y)\|_1 = 0$, $\mathbb{P}[a_s^{t+1} = a_c^{t+1} = 0 | N^t = C] \geq 1 - 2e^{-\gamma/2}$. By a union bound, recalling that X^t is fixed for all t by assumption, $\mathbb{P}[N^{t+1} = N^t | N^t = C] \geq 1 - (n+2)e^{-\gamma/2}$.

Case 2 : $\|C(Y)\|_1 = 1$

In this case by Definition 5.3.11, we have $C(y_i) \leq C(x_i)$ for all i, $C(a_s) = 1$, and $C(a_c) = 0$. We can thus apply Lemma 5.3.9, giving that $\mathbb{P}[Y^{t+1} = Y^t | N^t = C] \geq 1 - ne^{-\gamma/2}$. Additionally, by Lemma 5.3.7 conclusion (2), since $\|C(Y)\|_1 = 1$, $\mathbb{P}[a_s^{t+1} = 1 \text{ and } a_c^{t+1} = 0 | N^t = C] \geq 1 - 2e^{-\gamma/2}$. So by a union bound,

$$\mathbb{P}[N^{t+1} = N^t | N^t = C] \geq 1 - (n+2)e^{-\gamma/2},$$

giving the result in this case and completing the lemma. □

Lemma 5.3.12 immediately implies a bound on the probability that $\mathcal{T}_{n,\gamma}$ remains in a valid WTA configuration for t_s consecutive times.

Corollary 5.3.13 (Stability of Valid WTA Configurations)**.** *Assume the input execution α_X of $\mathcal{T}_{n,\gamma}$ has X^t fixed for all t. For any time t and valid WTA configuration C with $C(X) = X^t$,*

$$\mathbb{P}[N^t = N^{t+1} = ... = N^{t+t_s} | N^t = C] \geq 1 - t_s(n+2)e^{-\gamma/2}.$$

Proof. Applying Lemma 5.3.12 for each time $t+1, ..., t+t_s$ in succession gives the result. □

5.3.4 Convergence to Good Configurations

With the stability bound of Corollary 5.3.13 in place, it remains to prove that $\mathcal{T}_{n,\gamma}$ converges quickly to a valid WTA configuration. We do this in two main steps:

Configuration Type	$C(y_i) \leq C(x_i)\ \forall i$?	$\|C(Y)\|_1$	$C(a_s)$	$C(a_c)$
Valid WTA (Def. 5.3.11)	✓	$\min(1, \|C(X)\|_1)$	1	0
Near-Valid WTA (Def. 5.3.15)	✓	$\min(1, \|C(X)\|_1)$	1	1
Valid k-WTA (Def. 5.3.14)	✓	$k \geq 2$	1	1
Reset (Def. 5.3.16)	–	–	0	0

Table 5.1: Summary of good configuration types (Definition 5.3.17), from which we show rapid convergence to a valid WTA configuration. We refer to the configuration types shaded in gray as *active configurations* (Definition 5.3.18).

1. In this Section, we define three additional good configuration types and show that all other network configurations converge to a good configuration within just two time steps with high probability.
2. In Sections 5.3.5 and 5.3.6 we show that, in turn, each of these good configurations rapidly converges to a valid WTA configuration with constant probability.

A high level illustration of our proof is shown in Figure 5-2.

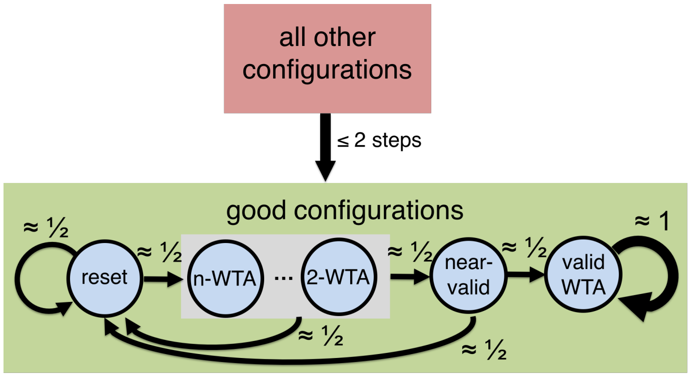

Figure 5-2: A high level illustration of our proof that $\mathcal{T}_{n,\gamma}$ solves the WTA problem. We show that all configurations converge to the set of good configurations. Once in a good configuration, $\mathcal{T}_{n,\gamma}$ converges to a valid WTA state very rapidly, with constant probability. In the illustration, arrow size corresponds to relative probability.

The first class of good configurations are *valid k-WTA configurations*. In such configurations the network behaves as expected before convergence. Multiple outputs corresponding to firing inputs fire and the inhibitors a_s and a_c fire, driving convergence towards a valid WTA configuration.

Definition 5.3.14 (Valid k-WTA Configuration). *For any $k \geq 2$, a* valid k-WTA configuration *of $\mathcal{T}_{n,\gamma}$ is a configuration C with $C(y_i) \leq C(x_i)$ for all $i \in \{1, ..., n\}$, $\|C(Y)\|_1 = k$ and $C(a_s) = C(a_c) = 1$.*

We next define a class of *near-valid WTA configurations*, each of which is a small perturbation of a valid WTA configuration, with correct output but incorrect inhibitor behavior. We will show in Section 5.3.6 that the network rapidly converges to a near-valid WTA configuration from any configuration. In turn, it transitions with probability $\approx 1/2$ from a near-valid WTA configuration to a valid WTA configuration.

Definition 5.3.15 (Near-Valid WTA Configuration)**.** *A* near-valid WTA configuration *of $\mathcal{T}_{n,\gamma}$ is a configuration C in which $C(Y)$ is a valid WTA output configuration for $C(X)$ (Definition 5.2.6) but $C(a_s) = C(a_c) = 1$.*

Finally, we define the class of *reset configurations* with $C(a_s) = C(a_c) = 0$. Since there is no inhibition in such a configuration, each output corresponding to a firing input will fire with probability $\geq 1/2$ at the next time. With probability $\approx 1/2$, the network will transition to either a valid k-WTA, a near-valid WTA, or a valid WTA configuration within three steps (Lemma 5.3.22).

Definition 5.3.16 (Reset Configuration)**.** *A* reset configuration *of $\mathcal{T}_{n,\gamma}$ is a configuration C with $C(a_s) = C(a_c) = 0$.*

Definition 5.3.17 (Good Configuration)**.** *A* good configuration *is any configuration that is either a valid WTA configuration (Definition 5.3.11), a valid k-WTA configuration (Definition 5.3.14), a reset configuration (Definition 5.3.16), or a near-valid WTA configuration (Definition 5.3.15).*

For conciseness, we also give a name to the good configurations excluding reset configurations:

Definition 5.3.18 (Active Configuration)**.** *An* active configuration *is any configuration that is either a valid WTA configuration (Definition 5.3.11), a valid k-WTA configuration (Definition 5.3.14), or a near-valid WTA configuration (Definition 5.3.15).*

We first give a simple lemma building on Lemma 5.3.9, which characterizes the network's behavior when a single inhibitor and at least one output corresponding to a firing input fire at time t:

Lemma 5.3.19. *Assume the input execution α_X of $\mathcal{T}_{n,\gamma}$ has X^t fixed for all t. For any time t and any configuration C with $C(X) = X^t$, ($C(a_s) = 1$ and $C(a_c) = 0$) or ($C(a_s) = 0$ and $C(a_c) = 1$), $\|C(Y)\|_1 \geq 1$, and $C(y_i) \leq C(x_i)$ for all i:*

$$\mathbb{P}[N^{t+1} \textit{ is a valid WTA or valid } k\textit{-WTA configuration } | N^t = C] \geq 1 - (n+2)e^{-\gamma/2}.$$

Proof. If $\|C(Y)\|_1 = 1$, then C is a valid WTA configuration. Thus, by Lemma 5.3.12, conditioned in $N^t = C$, N^{t+1} is a valid WTA configuration with probability $\geq 1 - (n+2)e^{-\gamma/2}$.

If $\|C(Y)\|_1 \geq 2$, then by Lemma 5.3.7 conclusion (3),

$$\mathbb{P}[a_s^{t+1} = a_c^{t+1} = 1 | N^t = C] \geq 1 - 2e^{-\gamma/2}.$$

Further, by Lemma 5.3.9, $\mathbb{P}[Y^{t+1} = Y^t | N^t = C] \geq 1 - ne^{-\gamma/2}$, which gives that N^{t+1} is a valid k-WTA configuration since, by conditioned on $N^t = C$, $\|Y^t\|_1 \geq 2$ and $y_i^t \leq x_i^t$ for all i. Thus, by a union bound, conditioned on $N^t = C$, N^{t+1} is a valid k-WTA configuration with probability $\geq 1 - (n+2)e^{-\gamma/2}$, giving the lemma.

□

Theorem 5.3.20 (Convergence to a Good Configuration)**.** *Assume that the input execution α_X of $\mathcal{T}_{n,\gamma}$ has X^t fixed for all t. For any time t and configuration C with $C(X) = X^t$,*

$$\mathbb{P}[\textit{at least one of } \{N^{t+1}, N^{t+2}\} \textit{ is a good configuration } | N^t = C] \geq 1 - 2(n+1)e^{-\gamma/2}.$$

Proof. Let $\mathcal{E}$ be the event that at least one of $\{N^{t+1}, N^{t+2}\}$ is a good configuration. Let $\mathcal{E}_1$ be the event that, $N^t = C$ and for all i, $y_i^{t+1} \leq x_i^{t+1}$. By Corollary 5.3.6 and the fact that X^t is fixed

$$\mathbb{P}[\mathcal{E}_1 | N^t = C] \geq 1 - ne^{-\gamma/2}. \tag{5.11}$$

We now give a simple case analysis, considering all values of a_c^{t+1} and a_s^{t+1}.

Case 1: $a_c^{t+1} = a_s^{t+1} = 0$**.**

In this case, N^{t+1} is a reset configuration (Definition 5.3.16). So we have:

$$\mathbb{P}\left[\mathcal{E} \middle| \mathcal{E}_1, a_c^{t+1} = a_s^{t+1} = 0\right] = 1. \tag{5.12}$$

Case 2: $a_c^{t+1} = a_s^{t+1} = 1$**.**

If $\|Y^{t+1}\|_1 = 0$, then N^{t+2} is a reset configuration with probability $\geq 1 - 2e^{-\gamma/2}$ by Lemma 5.3.7 conclusion (1). If $\|Y^{t+1}\|_1 = 1$ and $\mathcal{E}_1$ occurs, then N^{t+1} is a near-valid WTA configuration (Definition 5.3.15). If $\|Y^{t+1}\|_1 \geq 2$ and $\mathcal{E}_1$ occurs, then N^{t+1} is a valid k-WTA configuration (Definition 5.3.14). Thus:

$$\mathbb{P}\left[\mathcal{E} \middle| \mathcal{E}_1, a_c^{t+1} = a_s^{t+1} = 1\right] \geq 1 - 2e^{-\gamma/2}. \tag{5.13}$$

Case 3: ($a_s^{t+1} = 1$ and $a_c^{t+1} = 0$) or ($a_s^{t+1} = 0$ and $a_c^{t+1} = 1$).

Let $\mathcal{E}_2$ be the event that ($a_s^{t+1} = 1$ and $a_c^{t+1} = 0$) or ($a_s^{t+1} = 0$ and $a_c^{t+1} = 1$). Again, if $\|Y^{t+1}\|_1 = 0$, then N^{t+2} is a reset configuration with probability $\geq 1 - 2e^{-\gamma/2}$ by Lemma 5.3.7 conclusion (1).

If $\mathcal{E}_2$ and $\mathcal{E}_1$ occur and $\|Y^{t+1}\|_1 \geq 1$, we can apply Lemma 5.3.19, giving that N^{t+2} is either a valid WTA or valid k-WTA configuration with probability $\geq 1-(n+2)e^{-\gamma/2}$. We thus have:

$$\mathbb{P}[\mathcal{E}|\mathcal{E}_1, \mathcal{E}_2] \geq 1 - (n+2)e^{-\gamma/2}. \tag{5.14}$$

Completing the lemma.

Combining (5.12), (5.13), and (5.14), by the law of total probability, $\mathbb{P}[\mathcal{E}|\mathcal{E}_1] \geq 1 - (n+2)e^{-\gamma/2}$. We can then use that $\mathbb{P}[\mathcal{E}_1|N^t = C] \geq 1 - ne^{-\gamma/2}$ by (5.11) to give:

$$\begin{aligned}
\mathbb{P}[\mathcal{E}|N^t = C] &\geq \mathbb{P}[\mathcal{E}_1|N^t = C] \cdot \mathbb{P}[\mathcal{E}|\mathcal{E}_1, N^t = C] \\
&= \mathbb{P}[\mathcal{E}_1|N^t = C] \cdot \mathbb{P}[\mathcal{E}|\mathcal{E}_1] \quad \text{(Since, by definition, } \mathcal{E}_1 \text{ implies } N^t = C.\text{)} \\
&\geq 1 - 2(n+1)e^{-\gamma/2},
\end{aligned}$$

completing the lemma. □

5.3.5 Transition Lemmas for Good Configurations

We now give a set of lemmas that characterize the transitions of $\mathcal{T}_{n,\gamma}$ when starting from a good configuration. We first show that a near-valid WTA configuration transitions with probability ≈ 1/2 to the adjacent valid WTA configuration (i.e., the configuration with the same output behavior, but correct inhibitor behavior).

Lemma 5.3.21 (From Near-Valid to Valid Configurations)**.** *Assume that the input execution α_X of $\mathcal{T}_{n,\gamma}$ has X^t fixed for all t. For any time t and near-valid WTA configuration C with $C(X) = X^t$,*

$$\mathbb{P}[N^{t+1} \textit{ is a valid WTA configuration } |N^t = C] \geq 1/2 - (n+2)e^{-\gamma/2}.$$

Proof. We give a proof similar to that of Lemmas 5.3.9 and 5.3.12. We consider two cases:

Case 1: $\|C(Y)\|_1 = 0$.

In this case, for C to be a near-valid WTA configuration, according to Definition 5.3.15, we must also have $\|C(X)\|_1 = 0$. So by Corollary 5.3.6,

$$\mathbb{P}[\|Y^{t+1}\|_1 = 0 | N^t = C] \geq 1 - ne^{-\gamma/2}.$$

Additionally, by Lemma 5.3.7 conclusion (1), since $\|C(Y)\|_1 = 0$, $\mathbb{P}[a_s^{t+1} = a_c^{t+1} = 0 | N^t = C] \geq 1 - 2e^{-\gamma/2}$. By a union bound,

$$\mathbb{P}[\|N^{t+1}\|_1 = 0 | N^t = C] \geq 1 - (n+2)e^{-\gamma/2}.$$

By Definition 5.3.11, since $\|X^{t+1}\|_1 = 0$, if $\|N^{t+1}\|_1 = 0$, then N^{t+1} is a valid WTA configuration. So, conditioned on $N^t = C$, N^{t+1} is a valid WTA configuration with probability $\geq 1 - (n+2)e^{-\gamma/2}$, giving the lemma in this case.

Case 2: $\|C(Y)\|_1 = 1$.

In this case, for some i, $C(y_i) = 1$ and $C(y_j) = 0$ for all $j \neq i$. Further, we must have $C(x_i) = 1$ and $C(a_s) = C(a_c) = 1$ by the requirements of Definition 5.3.15. Define the event $\mathcal{E}_1$ by:

$$\mathcal{E}_1 \stackrel{\text{def}}{=} \left(y_i^{t+1} = 1,\ a_s^{t+1} = 1,\ a_c^{t+1} = 0 \text{ and } y_j^{t+1} = 0 \text{ for all } j \neq i \right).$$

By Lemma 5.3.10, $\mathbb{P}[y_i^{t+1} = 1 | N^t = C] = 1/2$ and $\mathbb{P}[y_i^t \leq y_i^{t+1} \text{ for all } i | N^t = C] \geq 1 - ne^{-\gamma/2}$. By Lemma 5.3.7 conclusion (2), since $\|C(Y)\|_1 = 1$, $\mathbb{P}[a_s^{t+1} = 1 \text{ and } a_c^{t+1} = 0 | N^t = C] \geq 1 - 2e^{-\gamma/2}$. By a union bound this gives,

$$\mathbb{P}[\mathcal{E}_1 | N^t = C] \geq 1/2 - (n+2)e^{-\gamma/2}.$$

If $\mathcal{E}_1$ occurs, then N^{t+1} is a valid WTA configuration, giving the lemma in this case. □

We next show that a reset configuration transitions to some active configuration (Defintion 5.3.18) with probability $\approx 1/2$.

Lemma 5.3.22 (From Reset to Active Configurations)**.** *Assume the input execution α_X of $\mathcal{T}_{n,\gamma}$ has X^t fixed for all t. For any time t and reset configuration C with $C(X) = C^t$,*

$$\mathbb{P}[\textit{at least one of } \{N^{t+1}, N^{t+2}, N^{t+3}\} \textit{ is active } | N^t = C] \geq 1/2 - 3(n+2)e^{-\gamma/2}.$$

Proof. Let $\mathcal{E}$ be the event that at least one of $\{N^{t+1}, N^{t+2}, N^{t+3}\}$ is an active configuration. We consider two cases:

Case 1: $\|X^t\|_1 = 0$.

Let $\mathcal{E}_{20}$ be the event that $\|N^{t+2}\|_1 = 0$. That is, that no neurons fire at time $t+2$. By Lemma 5.3.8, since $\|X^t\|_1 = 0$, $\mathbb{P}[\mathcal{E}_{20} | N^t = C] \geq 1 - 2(n+1)e^{-\gamma/2}$.

$\mathcal{E}_{20}$ requires that no neurons fire in N^{t+2}, which makes this a valid WTA configuration since $\|X^t\|_1 = 0$ and the input is fixed for all t. Thus, $\mathbb{P}[\mathcal{E} | N^t = C] \geq \mathbb{P}[\mathcal{E}_{20} | N^t = C] \geq 1 - 2(n+1)e^{-\gamma/2}$, completing the lemma in this case.

Case 2: $\|X^t\|_1 \geq 1$.

Let $\mathcal{E}_{11}$ be the event that $N^t = C$ and $\|Y^{t+1}\| \geq 1$ (i.e., at least one output fires at time $t+1$) and $y_i^{t+1} \leq x_i^{t+1}$ for all i. For any y_i with $x_i^t = 1$, we have:

$$\begin{aligned}\text{pot}(y_i, t+1) &= w(x_i, y_i)x_i^t + w(y_i, y_i)y_i^t + w(a_s, y_i)a_s^t + w(a_c, y_i)a_c^t - b(y_i)\\ &\geq 3\gamma + 0 + 0 + 0 - 3\gamma = 0.\end{aligned}$$

So, by Lemma 5.3.4, each output y_i with $x_i^t = 1$ fires with probability at least $1/2$ at time $t+1$. Each with $x_i^t = 0$ does not fire with probability $\geq 1 - e^{-\gamma/2}$ by Lemma 5.3.5. Since, by assumption, $\|X^t\|_1 \geq 1$, by a union bound, with probability $\geq 1/2 - ne^{-\gamma/2}$, at least one output with $x_i^t = 1$ fires in N^{t+1}, and no outputs with $x_i^t = 0$ fire. That is,

$$\mathbb{P}[\mathcal{E}_{11} | N^t = C] \geq 1/2 - ne^{-\gamma/2}. \tag{5.15}$$

We now proceed with a case analysis on the inhibitor behavior at time $t+1$:

Sub-case 1: $a_s^{t+1} = a_c^{t+1} = 1$.

In this case, assuming $\mathcal{E}_{11}$ occurs, N^{t+1} is either a valid k-WTA configuration for some k or a near-valid WTA configuration. We thus have:

$$\mathbb{P}\left[\mathcal{E}\middle|\mathcal{E}_{11}, a_s^{t+1} = a_c^{t+1} = 1\right] = 1. \tag{5.16}$$

Sub-case 2: $a_s^{t+1} = a_c^{t+1} = 0$.

Let $\mathcal{E}_{21}$ be the event that $\|Y^{t+2}\|_1 \geq 1$, $y_i^{t+2} \leq x_i^{t+2}$ for all i, and $a_s^{t+2} = 1$. Assuming $\mathcal{E}_{11}$ occurs and $a_s^{t+1} = a_c^{t+1} = 0$, any output with $y_i^{t+1} = 1$ has

$$\begin{aligned}\text{pot}(y_i, t+2) &= w(x_i, y_i)x_i^{t+1} + w(y_i, y_i)y_j^{t+1} + w(a_s, y_i)a_s^{t+1} + w(a_c, y_i)a_c^{t+1} - b(y_i)\\ &= 3\gamma + 2\gamma + 0 + 0 - 3\gamma > 0.\end{aligned}$$

Thus, any such output has $y_i^{t+2} = 1$ with probability $\geq 1 - e^{-\gamma/2}$ by Lemma 5.3.4. Combined with Corollary 5.3.5, with probability $\geq 1 - ne^{-\gamma/2}$, at least one output fires in N^{t+2}, and no outputs with $x_i^{t+1} = 0$ fire. Further by Lemma 5.3.7 conclusions (2) and (3), since $\mathcal{E}_{11}$ requires that $\|Y^{t+1}\| \geq 1$, with probability $\geq 1 - 2e^{-\gamma/2}$, $a_s^{t+2} = 1$. Thus,

$$\mathbb{P}\left[\mathcal{E}_{21}\middle|\mathcal{E}_{11}, a_s^{t+1} = a_c^{t+1} = 0\right] \geq 1 - (n+2)e^{-\gamma/2}. \tag{5.17}$$

Assume that both $\mathcal{E}_{11}$ and $\mathcal{E}_{21}$ occur. $\mathcal{E}_{21}$ requires that $a_s^{t+2} = 1$ and $\|Y^{t+2}\|_1 \geq 1$. If we also have $a_c^{t+2} = 1$, then N^{t+2} is a near-valid WTA or valid k-WTA configuration. If $a_c^{t+2} = 0$ and $\|Y^{t+2}\|_1 = 1$, then N^{t+2} is a valid WTA configuration. If $a_c^{t+2} = 0$ and $\|Y^{t+2}\|_1 \geq 2$, then by Lemma 5.3.7 conclusion (3), with probability $\geq 1 - 2e^{-\gamma/2}$, $a_s^{t+3} = a_c^{t+3} = 1$. Further, we can apply Lemma 5.3.9, giving that $Y^{t+3} = Y^{t+2}$ with probability $\geq 1 - ne^{-\gamma/2}$. This ensures that N^{t+3} is a valid k-WTA configuration. So we have by a union bound:

$$\mathbb{P}\left[\mathcal{E}\middle|\mathcal{E}_{11}, \mathcal{E}_{21}, a_s^{t+1} = a_c^{t+1} = 0\right] \geq 1 - (n+2)e^{-\gamma/2}.$$

Combined with (5.17) the above gives:

$$\begin{aligned}\mathbb{P}\left[\mathcal{E}\middle|\mathcal{E}_{11}, a_s^{t+1} = a_c^{t+1} = 0\right] &\geq \mathbb{P}\left[\mathcal{E}_{21}\middle|\mathcal{E}_{11}, a_s^{t+1} = a_c^{t+1} = 0\right] \cdot \mathbb{P}\left[\mathcal{E}\middle|\mathcal{E}_{11}, \mathcal{E}_{21}, a_s^{t+1} = a_c^{t+1} = 0\right]\\ &\geq \left(1 - (n+2)e^{-\gamma/2}\right) \cdot \left(1 - (n+2)e^{-\gamma/2}\right)\\ &\geq 1 - 2(n+2)e^{-\gamma/2}. \qquad (5.18)\end{aligned}$$

Sub-case 3: ($a_s^{t+1} = 1$ and $a_c^{t+1} = 0$) or ($a_s^{t+1} = 0$ and $a_c^{t+1} = 1$).

Let $\mathcal{E}_{13}$ denote the event that ($a_s^{t+1} = 1$ and $a_c^{t+1} = 0$) or ($a_s^{t+1} = 0$ and $a_c^{t+1} = 1$). Assuming $\mathcal{E}_{13}$ and $\mathcal{E}_{11}$, we can apply Lemma 5.3.19, which gives that N^{t+2} is either a valid WTA or valid k-WTA configuration with probability $\geq 1 - (n+2)e^{-\gamma/2}$. I.e.,

$$\mathbb{P}\left[\mathcal{E}\middle|\mathcal{E}_{11}, \mathcal{E}_{13}\right] \geq 1 - (n+2)e^{-\gamma/2}. \tag{5.19}$$

Completing Case 2.

Overall, combining (5.16), (5.18), and (5.19), by the law of total probability $\mathbb{P}[\mathcal{E}|\mathcal{E}_{11}] \geq 1 - 2(n+2)e^{-\gamma/2}$. Recalling that by (5.15) $\mathbb{P}[\mathcal{E}_{11}|N^t = C] \geq 1/2 - ne^{-\gamma/2}$:

$$\begin{aligned}
\mathbb{P}[\mathcal{E}|N^t = C] &\geq \mathbb{P}[\mathcal{E}_{11}|N^t = C] \cdot \mathbb{P}[\mathcal{E}|\mathcal{E}_{11}, N^t = C] \\
&= \mathbb{P}[\mathcal{E}_{11}|N^t = C] \cdot \mathbb{P}[\mathcal{E}|\mathcal{E}_{11}] \\
&\qquad\qquad \text{(Since, by assumption, } \mathcal{E}_{11} \text{ implies } N^t = C.\text{)} \\
&\geq 1/2 - 3(n+2)e^{-\gamma/2}
\end{aligned}$$

which completes the lemma in this case. □

We next show that, in each time step, with high probability, the number of firing outputs k does not increase. Further, with probability $\approx 1/2$, k is reduced by a factor of 1/2. This ensures rapid convergence towards having just a single firing output (i.e., a near-valid WTA state). While there is some chance that the convergence will 'overshoot' the target and zero outputs will fire at some time step, we show that the probability of this event is upper bounded by the probability of the desired event – i.e., reaching a near-valid WTA configuration.

Lemma 5.3.23 (Progress from k-WTA Configurations)**.** *Assume the input execution α_X of $\mathcal{T}_{n,\gamma}$ has X^t fixed for all t. For any time t and any valid k-WTA configuration C with $C(X) = X^t$,*

1. *Letting $\mathcal{E}$ be the event that N^{t+1} is either a near-valid WTA configuration, a valid k-WTA configuration with $\|Y^{t+1}\|_1 \leq \|Y^t\|_1$, or has $\|Y^{t+1}\|_1 = 0$.*

$$\mathbb{P}[\mathcal{E}|N^t = C] \geq 1 - (n+2)e^{-\gamma/2}.$$

2. $\mathbb{P}\left[\|Y^{t+1}\|_1 \leq \left\lceil \frac{\|Y^t\|_1}{2} \right\rceil \middle| N^t = C\right] \geq 1/2 - (n+2)e^{-\gamma/2}$.

3. $\mathbb{P}[\|Y^{t+1}\|_1 = 0|N^t = C] - (n+2)e^{-\gamma/2} \leq \mathbb{P}[N^{t+1}$ *is a near-valid* $|N^t = C]$.

Proof. Since C is a valid k-WTA configuration, conditioned on $N^t = C$, we have $\|Y^t\|_1 \geq 2$ and $a_s^t = a_c^t = 1$. Event $\mathcal{E}$ in (1) above occurs if and only if $a_s^{t+1} = a_c^{t+1} = 1$, and $y_i^{t+1} \leq y_i^t$ for all i.

By Lemma 5.3.7 conclusion (3), since $\|Y^t\|_1 \geq 2$, both inhibitors remain firing at time $t+1$ with high probability. That is,

$$\mathbb{P}[a_s^{t+1} = a_c^{t+1} = 1 | N^t = C] \geq 1 - 2e^{-\gamma/2}.$$

Further, by Lemma 5.3.10, $\mathbb{P}[y_i^{t+1} \leq y_i^t \text{ for all } i | N^t = C] \geq 1 - ne^{-\gamma/2}$. By a union bound, this gives that $\mathbb{P}[\mathcal{E}|N^t = C] \geq 1 - (n+2)e^{-\gamma/2}$, giving (1).

Let $\bar{Y} \subseteq Y$ denote the set of all output neurons with $y_i^t = 1$. Let $k = \|Y^t\|_1 = \|\bar{Y}^t\|_1$ and $k' = \|\bar{Y}^{t+1}\|_1$. By Lemma 5.3.10 properties (2) and (3), conditioned on $N^t = C$, k' is distributed according to the binomial distribution $B(k, 1/2)$. That is, it is the number of successes in k independent trials each with success probability $1/2$. Since $B(k, 1/2)$ is symmetric with mean $k/2$, its median is upper bounded by $\lceil k/2 \rceil$. Thus, $\mathbb{P}\left[k' \leq \lceil k/2 \rceil | N^t = C\right] \geq 1/2$. This gives by a union bound,

$$\mathbb{P}\left[k' \leq \lceil k/2 \rceil \text{ and } \mathcal{E} | N^t = C\right] \geq 1/2 - (n+2)e^{-\gamma/2}.$$

Note that if $\mathcal{E}$ holds, then $k' = \|\bar{Y}^{t+1}\|_1 = \|Y^{t+1}\|_1$, which thus gives conclusion (2).

Finally, we have $\mathbb{P}[k' = 1|N^t = C] = k \cdot \frac{1}{2^k}$ and $\mathbb{P}[k' = 0|N^t = C] = \frac{1}{2^k}$ and so

$$\mathbb{P}[k' = 0|N^t = C] \leq \mathbb{P}[k' = 1|N^t = C].$$

Let $\mathcal{E}_1$ be the event that N^{t+1} is a near-valid WTA configuration. Assuming $\mathcal{E}$ occurs, N^{t+1} is a near-valid WTA configuration if and only if $k' = 1$. That is,

$$\mathbb{P}[\mathcal{E}_1|\mathcal{E}, N^t = C] = \mathbb{P}[k' = 1|\mathcal{E}, N^t = C] = \mathbb{P}[k' = 1|N^t = C].$$

The second equality follows since $\mathcal{E}$ and k' are independent conditioned on N^t. k' only depends on the firing of $y \in \bar{Y}$ while $\mathcal{E}$ only depends on the firing of $u \in \{a_s, a_c\} \cup (Y \setminus \bar{Y})$. Using the above:

$$\begin{aligned}
\mathbb{P}[\mathcal{E}_1|N^t = C] &\geq \mathbb{P}[\mathcal{E}|N^t = C] \cdot \mathbb{P}[\mathcal{E}_1|\mathcal{E}, N^t = C] \\
&\geq \left(1 - (n+2)e^{-\gamma/2}\right) \cdot \mathbb{P}[k' = 1|N^t = C] \\
&\geq \mathbb{P}[k' = 0|N^t = C] - (n+2)e^{-\gamma/2} \\
&\geq \mathbb{P}[\|Y^{t+1}\|_1 = 0|N^t = C] - (n+2)e^{-\gamma/2}
\end{aligned}$$

where the last bound follows since $\|Y^{t+1}\|_1 \geq \|\bar{Y}_{t+1}\|_1 = k'$ so $\|Y^{t+1}\|_1 = 0$ at least requires $k' = 0$. This gives conclusion (3), completing the lemma. □

5.3.6 Convergence to WTA

We now use the good configuration transition probabilities given in Section 5.3.5, along with the results of Section 5.3.4, to show that, if sufficiently large γ, starting from any configuration $\mathcal{T}_{n,\gamma}$ converges with probability $\geq 1/18$ to a valid WTA configuration within $O(\log n)$ steps (see Lemma 5.3.32). The proof is in four main parts, which we outline here. We first define:

Definition 5.3.24 (Terminal Configuration)**.** *For $\mathcal{T}_{n,\gamma}$, a* terminal configuration *is any configuration C which is either a near-valid WTA configuration or has $\|C(Y)\|_1 = 0$ (i.e., no outputs fire).*

With this definition we can describe the general proof outline:

1. **Monotonicity** (Lemma 5.3.27). We prove that, starting from a k-WTA configuration, with high probability, $\mathcal{T}_{n,\gamma}$ remains in a k-WTA configuration, with the number of firing outputs consistently decreasing until it reaches a terminal configuration.

2. **Convergence** (Lemma 5.3.29). We prove that the number of firing outputs decreases rapidly. That is, starting from a k-WTA configuration, with high probability, a terminal configuration is reached within $O(\log n)$ steps.

3. **Probability of valid WTA** (Lemma 5.3.30, Corollary 5.3.31). We show that, starting from a valid k-WTA configuration, with constant probability, the terminal configuration reached is in fact a near-valid WTA configuration. By Lemma 5.3.21, with constant probability, this configuration transitions to a valid WTA configuration.

4. **Convergence from any starting configuration** (Theorem 5.3.32). We show that, starting in any configuration, with constant probability, the network reaches either a valid WTA configuration or a k-WTA configuration in few steps. Combined with our convergence results for k-WTA configurations, this proves fast convergence to a valid WTA state from any starting configuration.

We begin with a few definitions which we use to formalize the high level description above.

Definition 5.3.25 (Termination Step)**.** *Given any infinite execution* $\alpha = C^0C^1....$ *let* $\text{term}(\alpha, t, \Delta)$ *be the minimum value in* $\{t+1, ..., t+\Delta\}$ *for which* $C^{\text{term}(\alpha,t,\Delta)}$ *is a terminal configuration (Definition 5.3.24). If no such time exists let* $\text{term}(\alpha, t, \Delta) = t + \Delta$.

Definition 5.3.26 (Monotonicity Until Termination)**.** *Let* $\mathcal{E}_{mono}(t, \Delta)$ *be the event that the execution of* $\mathcal{T}_{n,\gamma}$ *is in set of executions* $\alpha = C^0C^1...$ *satisfying:*

$$\{\alpha | \textit{ for all } t' \in \{t+1, ..., \text{term}(\alpha, t, \Delta)\}, \ C^{t'} \textit{ is a valid } k\textit{-WTA configuration} \\ \textit{with } \|Y^{t'}\|_1 \leq \|Y^{t'-1}\|_1\}.$$

We begin by showing that, starting from any k-WTA configuration, with high probability $\mathcal{T}_{n,\gamma}$ behaves monotonically as described above.

Lemma 5.3.27 (Monotonicity)**.** *Assume the input execution* α_X *of* $\mathcal{T}_{n,\gamma}$ *has* X^t *fixed for all* t*. For any time* t*, any valid* k*-WTA configuration* C *with* $C(X) = X^t$*, and any* $\Delta \geq 1$*,*

$$\mathbb{P}[\mathcal{E}_{mono}(t, \Delta) | N^t = C] \geq 1 - \Delta(n+2)e^{-\gamma/2}.$$

Proof. Consider any $\Delta \geq 2$. If $\mathcal{E}_{mono}(t, \Delta - 1)$ occurs, then either $N^{t+(\Delta-1)}$ is a valid k-WTA configuration, or, for some $t' \in \{t+1, ..., t+\Delta-1\}$, $N^{t'}$ is a terminal configuration. Thus, by conclusion (1) of Lemma 5.3.23 we have:

$$\mathbb{P}[\mathcal{E}_{mono}(t, \Delta) | \mathcal{E}_{mono}(t, \Delta - 1), N^t = C] \geq 1 - (n+2)e^{-\gamma/2}. \tag{5.20}$$

Using (5.20) we can show by induction that for any $\Delta \geq 1$,

$$\mathbb{P}[\mathcal{E}_{mono}(t, \Delta) | N^t = C] \geq 1 - \Delta(n+2)e^{-\gamma/2}. \tag{5.21}$$

For any $\Delta \geq 2$, assume by way of induction that (5.21) holds for all $\Delta' < \Delta$. The assumption holds in the base case when $\Delta = 1$ again by conclusion (1) of Lemma 5.3.23, since C is a valid k-WTA configuration so $\mathbb{P}[\mathcal{E}_{mono}(t, 1) | N^t = C] \geq 1 - (n+$

$2)e^{-\gamma/2}$. Applying (5.20) and the inductive assumption:

$$\begin{aligned}\mathbb{P}[\mathcal{E}_{mono}(t,\Delta)|N^t = C] &\geq \mathbb{P}[\mathcal{E}_{mono}(t,\Delta-1)|N^t = C] \cdot \mathbb{P}[\mathcal{E}_{mono}(t,\Delta)|\mathcal{E}_{mono}(t,\Delta-1), N^t = C]\\ &\geq \left(1-(\Delta-1)(n+2)e^{-\gamma/2}\right)\cdot\left(1-(n+2)e^{-\gamma/2}\right)\\ &\geq 1-\Delta(n+2)e^{-\gamma/2}.\end{aligned}$$

which gives (5.21) for all $\Delta \geq 1$, and so the lemma. □

We next show that, starting from a k-WTA configuration, with high probability, $\mathcal{T}_{n,\gamma}$ reaches a terminal configuration within $O(\log n)$ steps. This requires showing that for $\Delta = O(\log n)$ with high probability, $N^{\text{term}(\alpha,t,\Delta)}$ (where $\text{term}(\alpha,t,\Delta)$ is defined in Definition 5.3.25) is actually a terminal configuration. We note that if the network does not reach a terminal configuration within Δ steps after time t, then, by definition, $N^{\text{term}(\alpha,t,\Delta)} = N^{t+\Delta}$, which is some non-terminal configuration.

We first define a termination event:

Definition 5.3.28 (Termination by Δ)**.** *Let $\mathcal{E}_{term}(t,\Delta)$ be the intersection of $\mathcal{E}_{mono}(t,\Delta)$ (Definition 5.3.26) and the event that $N^{\text{term}(\alpha,t,\Delta)}$ is a terminal configuration.*

Lemma 5.3.29 (Convergence from k-WTA Configurations)**.** *Assume the input execution α_X of $\mathcal{T}_{n,\gamma}$ has X^t fixed for all t and that $\gamma \geq 4\ln(n+2)+10$. Let $\Delta = 12(\log_2 n + 2)$. For any time t and valid k-WTA configuration C with $C(X) = X^t$,*

$$\mathbb{P}[\mathcal{E}_{term}(t,\Delta)|N^t = C] \geq 1-\Delta(n+2)e^{-\gamma/2} - \frac{1}{7n}$$

Proof. Let $\widehat{\mathcal{E}}_{term}(t,\Delta)$ be the event that $\mathcal{E}_{mono}(t,\Delta)$ occurs but $\mathcal{E}_{term}(t,\Delta)$ does not. For any $t' \in \{t+1,...,t+\Delta\}$, define the indicator $I_{t'} \in \{0,1\}$ with $I_{t'} = 1$ if and only if either:

- $N^{t'-1}$ is a valid k-WTA configuration and $\|Y^{t'}\|_1 \leq \lceil k/2 \rceil$.
- $N^{t'-1}$ is not a valid k-WTA configuration for any $k \geq 2$.

$\widehat{\mathcal{E}}_{term}(t,\Delta)$ requires that each of $N^t,...,N^{t+\Delta}$ is a valid k-WTA configuration and that

$$\|Y^t\|_1 \geq \|Y^{t+1}\|_1 \geq ... \geq \|Y^{t+\Delta}\|_1 \geq 2.$$

Otherwise, a terminal configuration with $\|Y^{t'}\|_1 = 0$ would be reached and $\mathcal{E}_{term}(t,\Delta)$ would occur.

Initially $\|Y^t\|_1 \leq n$. Since each time $I_{t'} = 1$, either $\|Y^t\|_1$ is cut in half or a configuration other than a valid k-WTA configuration occurs, $\widehat{\mathcal{E}}_{term}(t)$ can only occur if $\sum_{t'=t+1}^{t+\Delta} I_{t'} < \log_2 n + 1$. Thus we can bound:

$$\mathbb{P}[\widehat{\mathcal{E}}_{term}(t,\Delta)|N^t = C] \leq \mathbb{P}\left[\sum_{t'=t+1}^{t+\Delta} I_{t'} < (\log_2 n + 1)\big|N^t = C\right]. \tag{5.22}$$

We will show that this probability is low since $I_{t'} = 1$ with good probability. Specifically, if $N^{t'-1}$ is not a valid k-WTA configuration, then $I_{t'} = 1$ deterministically. If $N^{t'-1}$ is a valid k-WTA configuration, then by conclusion (2) of Lemma 5.3.23, $I_{t'} = 1$ with probability $\geq 1/2 - (n+2)e^{-\gamma/2}$. Overall, we have: $\mathbb{P}[I_{t'} = 1|N^{t'-1}] \geq 1/2 - (n+2)e^{-\gamma/2}$. In fact, by Lemma 5.2.2, we can also condition on all past configurations and have:

$$\mathbb{P}[I_{t'} = 1|N^{t'-1}N^{t'-2}...N^t, N^t = C] \geq 1/2 - (n+2)e^{-\gamma/2}.$$

The above bound lets us use Lemma 5.2.3 to upper bound the probability that $\sum_{t'=t+1}^{t+\Delta} I_{t'}$ is below any value d by the probability that a sum of Δ independent coin flips, each with success probability $1/2 - (n+2)e^{-\gamma/2}$, is below d. Specifically, let $Z_{t+1}, ..., Z_{t+\Delta}$ be i.i.d. random variables with $Z_{t'} = 1$ with probability $1/2 - (n+2)e^{-\gamma/2}$ and $Z_{t'} = 0$ otherwise. Invoking (5.22) and Lemma 5.2.3,

$$\mathbb{P}[\widehat{\mathcal{E}}_{term}(t,\Delta)|N^t = C] \leq \mathbb{P}\left[\sum_{t'=t+1}^{t+\Delta} I_{t'} < (\log_2 n + 1)\big|N^t = C\right] \tag{5.23}$$

$$\leq \mathbb{P}\left[\sum_{t'=t+1}^{t+\Delta} Z_{t'} < (\log_2 n + 1)\right]. \tag{5.24}$$

By our assumption that $\gamma \geq 4\ln(n+2) + 10$ and our setting of $\Delta = 12(\log_2 n + 2) \leq 14n$:

$$\mathbb{E}\left[\sum_{t'=t+1}^{t+\Delta} Z_{t'}\right] = \Delta/2 - \Delta(n+2)e^{-\gamma/2} \geq \Delta/3 = 4(\log_2 n + 2).$$

By a standard Chernoff bound [MU05],

$$\mathbb{P}\left[\sum_{t'=t+1}^{t+\Delta} Z_{t'} \leq (\log_2 n + 1)\right] \leq e^{-\frac{(3/4)^2 \cdot 4(\log_2 +2)}{2}} \leq e^{-(\log_2 n+2)} \leq \frac{1}{7n}.$$

We thus have, by (5.23), $\mathbb{P}[\widehat{\mathcal{E}}_{term}(t,\Delta)|N^t = C] \leq \frac{1}{7n}$. Combined with Lemma 5.3.27 this gives:

$$\begin{aligned}\mathbb{P}[\mathcal{E}_{term}(t,\Delta)|N^t = C] &= \mathbb{P}[\mathcal{E}_{mono}(t,\Delta)|N^t = C] - \mathbb{P}[\widehat{\mathcal{E}}_{term}(t,\Delta)|N^t = C] \\ &\geq 1 - \Delta(n+2)e^{-\gamma/2} - \frac{1}{7n}.\end{aligned}$$

□

We next combine Lemma 5.3.29 with conclusion (3) of Lemma 5.3.23 and Lemma 5.3.21 to show that, starting from a valid k-WTA configuration, not only does $\mathcal{T}_{n,\gamma}$ reach a terminal configuration quickly, but also, if γ is large enough, this terminal configuration is a near-valid WTA configuration with probability $\approx 1/2$.

Lemma 5.3.30 (Constant Probability of Near-Valid WTA, from k-WTA Configurations)**.** *Assume the input execution α_X of $\mathcal{T}_{n,\gamma}$ has X^t fixed for all t and that $\gamma \geq 4\ln(n+2) + 10$. Let $\Delta = 12(\log_2 n + 2)$ and $\mathcal{E}_1(t)$ be the event that there is some $t' \in \{t+1, ..., t+\Delta\}$, such that $N^{t'}$ is a near-valid WTA configuration. For any t and valid k-WTA configuration C with $C(X) = X^t$,*

$$\mathbb{P}[\mathcal{E}_1(t)|N^t = C] \geq \frac{1}{2} - \frac{\Delta+1}{2}(n+2)e^{-\gamma/2} - \frac{1}{14n}.$$

Proof. $\mathcal{E}_1(t)$ is equivalent to the event that $\mathcal{E}_{term}(t,\Delta)$ occurs and $N^{\text{term}(\alpha,t,\Delta)}$ is a near-valid WTA configuration. Let $\mathcal{E}_0(t)$ be the event that $\mathcal{E}_{term}(t,\Delta)$ occurs and $\|Y^{\text{term}(\alpha,t,\Delta)}\|_1 = 0$. $\mathcal{E}_0(t)$ and $\mathcal{E}_1(t)$ are disjoint with $\mathcal{E}_0(t) \cup \mathcal{E}_1(t) = \mathcal{E}_{term}(t,\Delta)$. So by Lemma 5.3.29,

$$\begin{aligned}\mathbb{P}[\mathcal{E}_0(t)|N^t = C] + \mathbb{P}[\mathcal{E}_1(t)|N^t = C] &= \mathbb{P}[\mathcal{E}_{term}(t,\Delta)|N^t = C] \\ &\geq 1 - \Delta(n+2)e^{-\gamma/2} - \frac{1}{7n}. \qquad (5.25)\end{aligned}$$

We will use conclusion (3) of Lemma 5.3.23 to show that

$$\mathbb{P}[\mathcal{E}_1(t)|N^t = C] \geq \mathbb{P}[\mathcal{E}_0(t)|N^t = C] - (n+2)e^{-\gamma/2}, \qquad (5.26)$$

which combined with (5.25) gives the conclusion of the lemma, that

$$\mathbb{P}[\mathcal{E}_1(t)|N^t = C] \geq \frac{1}{2} - \frac{\Delta+1}{2}(n+2)e^{-\gamma/2} - \frac{1}{14n}. \qquad (5.27)$$

For each $t' \in \{t+1, ..., t+\Delta\}$, let $\mathcal{E}_{term}(t,t',\Delta)$ be the event that $\mathcal{E}_{term}(t,\Delta)$ occurs and $\text{term}(\alpha,t,\Delta) = t'$. Define $\mathcal{E}_0(t,t')$ and $\mathcal{E}_1(t,t')$ analogously. Let $\mathcal{E}_2(t,t')$ be the event that $N^t, ..., N^{t'-1}$ are all valid k-WTA configurations with $\|Y^t\|_1 \geq ... \geq \|Y^{t'-1}\|_1 \geq 2$. Let $\bar{\mathcal{E}}_2(t,t')$ be its complement.

$$\mathbb{P}[\mathcal{E}_1(t,t')|\bar{\mathcal{E}}_2(t,t')] = \mathbb{P}[\mathcal{E}_0(t,t')|\bar{\mathcal{E}}_2(t,t')] = 0 \tag{5.28}$$

since both $\mathcal{E}_1(t,t')$ and $\mathcal{E}_0(t,t')$ require $\mathcal{E}_{mono}(t,\Delta)$ to hold, which requires $\mathcal{E}_2(t,t')$ to hold if $\text{term}(\alpha,t,\Delta) = t'$. Further, by conclusion (3) of Lemma 5.3.23, since $\mathcal{E}_2(t,t')$ requires that $N^{t'-1}$ is a valid k-WTA configuration,

$$\mathbb{P}[\mathcal{E}_1(t,t')|\mathcal{E}_2(t,t'), N^t = C] \geq \mathbb{P}[\mathcal{E}_0(t,t')|\mathcal{E}_2(t,t'), N^t = C] - (n+2)e^{-\gamma/2} \tag{5.29}$$

By the law of total probability, (5.28) and (5.29) give

$$\mathbb{P}[\mathcal{E}_1(t,t')|N^t = C] \geq \mathbb{P}[\mathcal{E}_0(t,t')|N^t = C] - (n+2)e^{-\gamma/2}.$$

Again by the law of total probability, this gives

$$\mathbb{P}[\mathcal{E}_1(t)|N^t = C] \geq \mathbb{P}[\mathcal{E}_0(t)|N^t = C] - (n+2)e^{-\gamma/2},$$

yielding (5.26) and thus (5.27) and the lemma. □

We next combine Lemma 5.3.30 with Lemma 5.3.21, which shows that any near-valid WTA configuration transitions with probability $\approx 1/2$ to a valid WTA configuration. This gives fast convergence to a valid WTA configuration starting from any valid k-WTA configuration, with probability $\geq 1/8$.

Corollary 5.3.31 (Constant Probability of Success, from k-WTA Configurations)**.** *Assume the input execution α_X of $\mathcal{T}_{n,\gamma}$ has X^t fixed for all t and that $\gamma \geq 4\ln(n+2)+10$. Let $\mathcal{E}(t)$ be the event that there is some $t' \in \{t+1, ..., t+12\log_2 n+25\}$, such that $N^{t'}$ is a valid WTA configuration. For any t and valid k-WTA configuration C with $C(X) = X^t$,*

$$\mathbb{P}[\mathcal{E}(t)|N^t = C] \geq 1/8.$$

Proof. As in Lemma 5.3.30, let $\Delta = 12(\log_2 n + 2)$ and $\mathcal{E}_1(t)$ be the event that there is some $t' \in \{t+1, ..., t+\Delta\}$, such that $N^{t'}$ is a near-valid WTA configuration. Let $\mathcal{E}_{val}(t)$ be the event that $N^{t'+1}$ is a valid WTA configuration. We have $\mathcal{E}(t) \subseteq \mathcal{E}_{val}(t)$

(since $t' + 1 \in \{t+2, ..., t+\Delta+1\}$ where $t + \Delta + 1 = t + (12\log_2 n + 25)$). Thus it suffices to show that $\mathbb{P}[\mathcal{E}_{val}(t)|N^t = C] \geq 1/8$.

By Lemmas 5.3.30 and 5.3.21, for any configuration C:

$$\begin{aligned}\mathbb{P}[\mathcal{E}_{val}(t)|N^t = C] &\geq \mathbb{P}[\mathcal{E}_1(t)|N^t = C] \cdot \mathbb{P}[\mathcal{E}_{val}(t)|\mathcal{E}_1(t), N^t = C]\\ &\geq \left(\frac{1}{2} - \frac{\Delta+1}{2}(n+2)e^{-\gamma/2} - \frac{1}{14n}\right) \cdot \left(1/2 - (n+2)e^{-\gamma/2}\right)\\ &\geq \frac{1}{4} - \frac{\Delta+3}{4}(n+2)e^{-\gamma/2} - \frac{1}{28n}.\end{aligned}$$

We can loosely bound $\frac{\Delta+3}{4} = \frac{12(\log_2 n+2)+3}{4} \leq \frac{12(n+2n)+3n}{4} \leq 10n$. Further, by our assumption that $\gamma \geq 4\ln(n+2) + 10$ we have:

$$\mathbb{P}[\mathcal{E}_{val}(t)|N^t = C] \geq \frac{1}{4} - \frac{10n(n+2)}{(n+2)^2 \cdot e^5} - \frac{1}{28} \geq \frac{1}{8}$$

which gives the corollary. □

Finally, we show that, starting from *any configuration*, with constant probability, $\mathcal{T}_{n,\gamma}$ converges to a valid WTA configuration in $O(\log n)$ steps. Our proof combines Theorem 5.3.20 and Lemma 5.3.22 which show that any configuration transitions to an active configuration (Definition 5.3.18) in few steps with constant probability. We then apply Corollary 5.3.31 to show convergence from such a configuration.

Theorem 5.3.32. *Assume the input execution α_X of $\mathcal{T}_{n,\gamma}$ has X^t fixed for all t and that $\gamma \geq 4\ln(n+2) + 10$. Let $\mathcal{E}(t)$ be the event that there is some $t' \in \{t+1, ..., t + (12\log_2 n + 30)\}$ such that $N^{t'}$ is a valid WTA configuration. For any time t and configuration C with $C(X) = X^t$,*

$$\mathbb{P}[\mathcal{E}(t)|N^t = C] \geq 1/18.$$

Proof. Let $\mathcal{E}_{act}(t)$ be the event that $N^t = C$ and that at least one of $\{N^{t+1}, ..., N^{t+5}\}$ is an active configuration (Definition 5.3.18). Let $N^{\hat{t}}$ be the first active configuration in this set, or $N^{\hat{t}} = N^{t+5}$ if there is no such configuration.

By Theorem 5.3.20, conditioned on $N^t = C$, with probability $\geq 1 - 2(n+1)e^{-\gamma/2}$ one of $\{N^{t+1}, N^{t+2}\}$ is a good configuration. Let $N^{\tilde{t}}$ be the first good configuration in this set or $N^{\tilde{t}} = N^{t+2}$ if neither are good. If $N^{\tilde{t}}$ is also an active configuration then $\mathcal{E}_{act}(t)$ holds.

If not, then $N^{\tilde{t}}$ is a reset configuration. Let C' be any reset configuration. By Lemma 5.3.22, conditioned on $N^{\tilde{t}} = C'$, with probability $\geq 1/2 - 3(n+2)e^{-\gamma/2}$ at least one of $\{N^{\tilde{t}+1}, N^{\tilde{t}+2}, N^{\tilde{t}+3}\}$, is an active configuration. Thus, overall we have:

$$\begin{aligned}\mathbb{P}[\mathcal{E}_{act}|N^t = C] &\geq \left(1 - 2(n+1)e^{-\gamma/2}\right) \cdot \left(1/2 - 3(n+2)e^{-\gamma/2}\right)\\ &\geq \frac{1}{2} - 5(n+2)e^{-\gamma/2}. \end{aligned} \tag{5.30}$$

We can define three disjoint events:

$$\begin{aligned}\mathcal{E}_{act,1}(t) &\overset{\text{def}}{=} (N^{\hat{t}} \text{ is a valid } k\text{-WTA configuration })\\ \mathcal{E}_{act,2}(t) &\overset{\text{def}}{=} (N^{\hat{t}} \text{ is a near-valid WTA configuration })\\ \mathcal{E}_{act,3}(t) &\overset{\text{def}}{=} (N^{\hat{t}} \text{ is a valid WTA configuration })\end{aligned}$$

We have $\mathcal{E}_{act}(t) = \bigcup_{i=1}^{3} \mathcal{E}_{act,i}(t)$ and so by the law of total probability:

$$\begin{aligned}\mathbb{P}[\mathcal{E}(t)|\mathcal{E}_{act}(t)] &= \sum_{i=1}^{3} \mathbb{P}[\mathcal{E}(t)|\mathcal{E}_{act,i}(t)] \cdot \mathbb{P}[\mathcal{E}_{act,i}(t)|\mathcal{E}_{act}(t)]\\ &\geq \sum_{i=1}^{3} \min_{i\in\{1,2,3\}} \mathbb{P}[\mathcal{E}(t)|\mathcal{E}_{act,i}(t)] \cdot \mathbb{P}[\mathcal{E}_{act,i}(t)|\mathcal{E}_{act}(t)]\\ &\geq \min_{i\in\{1,2,3\}} \cdot \mathbb{P}[\mathcal{E}(t)|\mathcal{E}_{act,i}(t)] \end{aligned} \tag{5.31}$$

where the last bound follows since the $\mathcal{E}_{act,i}(t)$ events are disjoint and $\sum_{i=1}^{3} \mathbb{P}[\mathcal{E}_{act,i}(t)|\mathcal{E}_{act}(t)] = 1$. We now bound this minimum via a case analysis:

Case 1: $\mathbb{P}[\mathcal{E}(t)|\mathcal{E}_{act,1}(t)]$

In this case, $N^{\hat{t}}$ is a valid k-WTA configuration, so applying Corollary 5.3.31, conditioned on $\mathcal{E}_{act,1}(t)$, with probability $1/8$ there is some time $t' \in \{\hat{t}+1, ..., \hat{t} + (12\log_2 n + 25)\}$ such that $N^{t'}$ is a valid WTA configuration. Note that $\hat{t} \leq t+5$ giving $t' \leq (12 \log_2 n + 30)$. We thus have:

$$\mathbb{P}[\mathcal{E}(t)|\mathcal{E}_{act,1}(t)] \geq 1/8. \tag{5.32}$$

Case 2: $\mathbb{P}[\mathcal{E}(t)|\mathcal{E}_{act,2}(t)]$

In this case, $N^{\hat{t}}$ is a near-valid WTA configuration, so applying Lemma 5.3.21, conditioned on $\mathcal{E}_{act,2}(t)$, $N^{\hat{t}+1}$ is a valid WTA configuration with probability $\geq 1/2 -$

$(n+2)e^{-\gamma/2}$. By our assumption that $\gamma \geq 4\ln(n+2)+10$:

$$\mathbb{P}[\mathcal{E}(t)|\mathcal{E}_{act,2}(t)] \geq 1/2 - (n+2)e^{-\gamma/2} \geq 1/3. \tag{5.33}$$

Case 3: $\mathbb{P}[\mathcal{E}(t)|\mathcal{E}_{act,3}(t)]$

If $\mathcal{E}_{act,3}(t)$ holds then $N^{\hat{t}}$ is a valid WTA configuration by definition, so trivially

$$\mathbb{P}[\mathcal{E}(t)|\mathcal{E}_{act,3}(t)] = 1. \tag{5.34}$$

Completing the theorem:

Combining (5.32), (5.33), (5.34), and (5.31) we have $\mathbb{P}[\mathcal{E}(t)|\mathcal{E}_{act}(t)] \geq 1/8$. Using (5.30) we then have:

$$\begin{aligned}
\mathbb{P}[\mathcal{E}(t)|N^t = C] &\geq \mathbb{P}[\mathcal{E}_{act}(t)|N^t = C] \cdot \mathbb{P}[\mathcal{E}(t)|\mathcal{E}_{act}(t), N^t = C] \\
&= \mathbb{P}[\mathcal{E}_{act}(t)|N^t = C] \cdot \mathbb{P}[\mathcal{E}(t)|\mathcal{E}_{act}(t)] \\
&\qquad \text{(Since } \mathcal{E}_{act}(t) \subseteq (N^t = C) \text{ by definition.)} \\
&\geq \frac{1}{8} \cdot \left(\frac{1}{2} - 5(n+2)e^{-\gamma/2}\right) \geq \frac{1}{18}
\end{aligned}$$

where the last bound follows from our assumption that $\gamma \geq 4\ln(n+2)+10$. □

5.3.7 Completing the Bounds

Given Theorem 5.3.32, it is easy to show that, with γ set large enough, $\mathcal{T}_{n,\gamma}$ solves the WTA problem (Definitions 5.2.7 and 5.2.9), giving Theorem 5.3.2 and 5.3.3. We start with the basic WTA problem of Definition 5.2.7. By Theorem 5.3.32, starting from any configuration, the network converges to a valid WTA configuration in $O(\log n)$ steps. By applying this analysis in sequence to $O(\log 1/\delta)$ sets of $O(\log n)$ steps, we show that the network converges to a valid WTA state with probability $\geq 1-\delta$ within $O(\log n \cdot \log(1/\delta))$ steps. Further, if γ is large enough, by Lemma 5.3.13, it remains in this state for t_s steps with high probability.

Theorem 5.3.2 (Two-Inhibitor WTA). *For $\gamma \geq 4\ln((n+2)t_s/\delta)+10$, $\mathcal{T}_{n,\gamma}$ solves* WTA(n, t_c, t_s, δ) *for any $t_c \geq 72(\log_2 n + 1) \cdot (\log_2(1/\delta)+1)$.*

Proof. Consider $\mathcal{T}_{n,\gamma}$ starting from any initial configuration N^0 and given an infinite input execution α_X with X^t fixed for all t. Let $\Delta = (12\log_2 n + 30)$ and

$r = 6(\log_2(1/\delta)+1)$. Let $\mathcal{E}$ be the event that there is some time $t \leq t_c$ where N^t is a valid WTA configuration.

For any $i \geq 0$, let $\mathcal{E}_i$ be the event that there is some time $t \in \{i\Delta+1, ..., (i+1)\Delta\}$ where N^t is a valid WTA configuration. By Theorem 5.3.32 and Lemma 5.2.2 we have:

$$\mathbb{P}[\mathcal{E}_i | N^{i\Delta}] = \mathbb{P}[\mathcal{E}_i | N^{i\Delta}, N^{i\Delta-1}, ..., N^1] \geq 1/8.$$

Let $Z_0, ..., Z_{r-1} \in \{0,1\}$ be independent coin flips, with $\mathbb{P}[Z_i = 1] = 1/8$. Applying Lemma 5.2.3:

$$\mathbb{P}[\mathcal{E}] = \mathbb{P}\left[\bigcap_{i=0}^{r-1} \mathcal{E}_i\right] \geq \mathbb{P}\left[\sum_{i=0}^{r-1} Z_i \geq 1\right] = 1 - \left(\frac{7}{8}\right)^r.$$

Using that $r = 6(\log_2(1/\delta)+1)$:

$$\mathbb{P}[\mathcal{E}] \geq 1 - \left(\frac{7}{8}\right)^{6(\log_2(1/\delta)+1)} \geq 1 - \frac{\delta}{2}.$$

Thus, with probability $\geq 1 - \frac{\delta}{2}$ there is some time $t \leq r \cdot \Delta \leq 72(\log_2 n + 1) \cdot (\log_2(1/\delta)+1) \leq t_c$ in which N^t is a valid WTA configuration. By Corollary 5.3.13, if C is a valid WTA configuration then

$$\mathbb{P}[N^t = N^{t+1} = ... = N^{t+t_s} | N^t = C] \geq 1 - t_s(n+2)e^{-\gamma/2} \geq 1 - \frac{\delta}{e^5},$$

where the bound holds by our assumption that $\gamma \geq 4\ln((n+2)t_s/\delta) + 10$. We thus have that the network reaches a valid WTA configuration within time t_c and remains in it for time t_s with probability $\geq \left(1 - \frac{\delta}{2}\right) \cdot \left(1 - \frac{\delta}{e^5}\right) \geq 1 - \delta$, yielding the theorem. □

We conclude by showing with what parameters $\mathcal{T}_{n,\gamma}$ solves the expected-time WTA problem of Definition 5.2.9.

Theorem 5.3.3 (Two-Inhibitor Expected-Time WTA). *For $\gamma \geq 4\ln((n+2)t_s)+10$, $\mathcal{T}_{n,\gamma}$ solves* WTA-EXP(n, t_c, t_s) *for any $t_c \geq 108(\log_2 n + 3)$.*

Proof. Recall that in Definition 5.2.9 we defined the convergence time for any infinite input execution α_X and output execution α_Y:

$$t(\alpha_X, t_s, \alpha_Y) = \min\left\{t : Y^t \text{ is a valid WTA output configuration for } X^t \text{ and } Y^t = ... = Y^{t+t_s}\right\}.$$

Define the worst case expected convergence time of $\mathcal{T}_{n,\gamma}$ on input α_X by:

$$t_{max}(\alpha_X) = \max_{N^0} \left(\mathop{\mathbb{E}}_{\alpha_Y \sim \mathcal{D}_Y(\mathcal{T}_{n,\gamma}, N^0, \alpha_X)} t(\alpha_X, t_s, \alpha_Y) \right).$$

To prove the lemma we must prove that for any α_X with X^t fixed for all t, $t_{max}(\alpha_X) \leq 108(\log_2 n + 3)$. Fixing such an α_X, for any starting configuration N^0, let $\mathop{\mathbb{E}}_{N_0}$ and $\mathop{\mathbb{P}}_{N_0}$ denote the expectation and probability of an event taken over executions drawn from $\mathcal{D}(\mathcal{T}_{n,\gamma}, N^0, \alpha_X)$.

Let $\Delta = (12 \log_2 n + 30)$ and let $\mathcal{E}_1$ be the event that there is some $t \in \{1, ..., \Delta\}$ where N^t is a valid WTA configuration. Let $\mathcal{E}_{stab}$ be the event that there is some $t \in \{1, ..., \Delta\}$ where N^t is a valid WTA configuration and additionally, where $N^t = ... = N^{t+t_s}$. Let $\bar{\mathcal{E}}_1$ and $\bar{\mathcal{E}}_{stab}$ be the complements of these two events. By Theorem 5.3.32, for any initial configuration N^0

$$\mathop{\mathbb{P}}_{N^0}[\mathcal{E}_1] \geq 1/8. \tag{5.35}$$

Further, by Corollary 5.3.13, if C is a valid WTA configuration then,

$$\mathop{\mathbb{P}}_{N^0}[N^t = N^{t+1} = ... = N^{t+t_s} | N^t = C] \geq 1 - t_s(n+2)e^{-\gamma/2} \geq 1 - \frac{1}{t_s \cdot e^5} \tag{5.36}$$

where the bound holds since $\gamma \geq 4\ln((n+2)t_s) + 10 \geq 2\ln((n+2)t_s^2) + 10$ and so $e^{-\gamma/2} \leq \frac{1}{(n+2)t_s^2 \cdot e^5}$. Together (5.35) and (5.36) give that:

$$\mathop{\mathbb{P}}_{N^0}[\mathcal{E}_{stab}] \geq \mathop{\mathbb{P}}_{N^0}[\mathcal{E}_{stab} | \mathcal{E}_1] \cdot \mathop{\mathbb{P}}_{N^0}[\mathcal{E}_1] \geq \frac{1}{8} \cdot \left(1 - \frac{1}{t_s \cdot e^5}\right) \geq \frac{1}{8} - \frac{1}{t_s \cdot e^5}.$$

We can write:

$$\begin{aligned}
\mathop{\mathbb{E}}_{N_0}[t(\alpha_X, t_s, \alpha_Y)] = &\mathop{\mathbb{E}}_{N_0}[t(\alpha_X, t_s, \alpha_Y) | \mathcal{E}_{stab}] \cdot \mathop{\mathbb{P}}_{N_0}[\mathcal{E}_{stab}] \\
&+ \mathop{\mathbb{E}}_{N_0}[t(\alpha_X, t_s, \alpha_Y) | \mathcal{E}_1, \bar{\mathcal{E}}_{stab}] \cdot \mathop{\mathbb{P}}_{N_0}[\mathcal{E}_1, \bar{\mathcal{E}}_{stab}] \\
&+ \mathop{\mathbb{E}}_{N_0}[t(\alpha_X, t_s, \alpha_Y) | \bar{\mathcal{E}}_1] \cdot \mathop{\mathbb{P}}_{N_0}[\bar{\mathcal{E}}_1]
\end{aligned} \tag{5.37}$$

Conditioned on $\mathcal{E}_{stab}$ (which also requires that $\mathcal{E}_1$ occurs), the network converges within Δ steps and stabilizes for t_s steps. Thus, we have:

$$\mathop{\mathbb{E}}_{N_0}[t(\alpha_X, t_s, \alpha_Y) | \mathcal{E}_{stab}] \leq \Delta.$$

Conditioned on $\mathcal{E}_1, \bar{\mathcal{E}}_{stab}$ the network converges, but does not stabilize. We can bound

$$\begin{aligned}\mathop{\mathbb{E}}_{N_0}[t(\alpha_X, t_s, \alpha_Y)|\mathcal{E}_1, \bar{\mathcal{E}}_{stab}] &\leq (\Delta + t_s) + \mathop{\mathbb{E}}_{N^{\Delta+t_s}}[t(\alpha_X, t_s, \alpha_Y)]\\ &\leq \Delta + t_s + t_{max}(\alpha_X).\end{aligned}$$

Finally, conditioned on $\bar{\mathcal{E}}_1$, the network does not converge within Δ steps. We have:

$$\begin{aligned}\mathop{\mathbb{E}}_{N_0}[t(\alpha_X, t_s, \alpha_Y)|\bar{\mathcal{E}}_1] &\leq \Delta + \mathop{\mathbb{E}}_{N^{\Delta}}[t(\alpha_X, t_s, \alpha_Y)]\\ &\leq \Delta + t_{max}(\alpha_X).\end{aligned}$$

We can plug these bounds along with the probability bounds of (5.35) and (5.36) into (5.37) to obtain:

$$\begin{aligned}\mathop{\mathbb{E}}_{N_0}[t(\alpha_X, t_s, \alpha_Y)] &\leq \Delta \cdot \left(\frac{1}{8} - \frac{1}{t_s \cdot e^5}\right) + (\Delta + t_{max}(\alpha_X) + t_s) \cdot \frac{1}{t_s \cdot e^5} + (\Delta + t_{max}(\alpha_X)) \cdot \frac{7}{8}\\ &\leq \Delta + t_{max}\left(\frac{7}{8} + \frac{1}{t_s \cdot e^5}\right) + \frac{t_s}{t_s \cdot e^5}\\ &\leq \Delta + t_{max}(\alpha_X) \cdot \frac{8}{9} + \frac{1}{e^5}.\end{aligned}$$

Since this bound holds for all N^0 we have:

$$t_{max}(\alpha_X) \leq \Delta + t_{max}(\alpha_X) \cdot \frac{8}{9} + \frac{1}{e^5}$$

which gives $t_{max}(\alpha_X) \leq 9\Delta + \frac{9}{e^5} \leq 9\Delta + 1$. This bound holds for all α_X and so gives the lemma, after recalling that $\Delta = (12\log_2 n + 30)$ so $9\Delta + 1 \leq 108(\log_2 n + 3)$. □

5.4 WTA Lower Bounds

The simple family of two-inhibitor networks presented in Section 5.3 gives convergence to a valid WTA output configuration in $O(\log n \cdot \log(1/\delta))$ steps with probability $\geq 1 - \delta$, as long as the weight scaling parameter γ is set large enough. Specifically, by Theorem 5.3.2, these networks solve WTA(n, t_c, t_s, δ) with $t_c = O(\log n \cdot \log(1/\delta))$ and t_s exponentially large in γ. In this section we ask whether this is optimal, considering two questions:

1. Are there networks that achieve comparable convergence speed with just a single auxiliary neuron?

2. Are there networks using two auxiliary neurons that converge faster?

We answer these questions for somewhat restricted classes of *simple SNNs* and *symmetric SNNs*, described in Definitions 5.4.1 and 5.4.2 below. These classes of networks include, in particular, the construction studied in Section 5.3.

We show that a simple SNN with just a single auxiliary neuron cannot solve WTA(n, t_c, t_s, δ) with $t_s = \tilde{\Omega}\left(\frac{t_c}{\log n}\right)$. That is, the network cannot effectively converge to a valid WTA output state and remain in this state for a significant time compared to its convergence time. Additionally, we show that no symmetric SNN with two auxiliary neurons can improve on the convergence time of the two-inhibitor network $\mathcal{T}_{n,\gamma}$ proven in Theorem 5.3.2 by more than a $O(\log \log n)$ factor.

We define the restricted network classes we consider in our lower bounds below.

Definition 5.4.1 (Simple SNN). *A spiking neural network* $\mathcal{N} = \langle N, w, b, f\rangle$ *is a* simple SNN *if it contains* n *input neurons labeled* $x_1, ..., x_n$ *and* n *output neurons labels* $y_1, ...y_n$ *and satisfies:*

- $w(x_i, y_j) = 0$ *and* $w(y_i, y_j) = 0$ *for all* $j \neq i$. *I.e., each input does not connect to outputs, other than its corresponding output, and outputs do not connect to each other.*

Note that in a simple SNN, auxiliary neurons may connect to each other, may have incoming edges from the input neurons, and may form unrestricted connections with the output neurons. In our two-auxiliary neuron lower bound we consider a further restricted class of networks:

Definition 5.4.2 (Symmetric SNN). *A simple SNN* $\mathcal{N} = \langle N, w, b, f\rangle$ *is a* symmetric SNN *if it contains* n *input neurons labeled* $x_1, ..., x_n$ *and* n *output neurons labels* $y_1, ...y_n$ *and satisfies:*

- *For all* $u, v \in A$, $w(u, v) = w(v, u) = 0$. *I.e., there are no connections between auxiliary neurons.*

- *For all* $u \in A$, $w(y_i, u) = w(y_j, u)$. *I.e., each auxiliary neuron is affected in the same way by each output.*

- *For all* i, j, $w(x_i, y_i) = w(x_j, y_j)$, $w(y_i, y_i) = w(y_j, y_j)$, $w(u, y_i) = w(u, y_j)$ *for all* $u \in A$, *and* $b(y_i) = b(y_j)$. *I.e., all outputs have identical incoming connections from their corresponding inputs, themselves, and the auxiliary neurons, and have identical biases.*

5.4.1 Single Auxiliary Neuron Lower Bound

We begin with our lower bound for simple SNNs with just a single auxiliary neuron.

Theorem 5.4.3 (One Neuron Lower Bound)**.** *For any $n \geq 20$, $\delta \leq 1/2$, and any spike probability function $f : \mathbb{R} \to [0,1]$ (satisfying the restrictions in Section 5.2.1), there is no simple SNN $\mathcal{N} = \langle N, w, b, f \rangle$ with just a single auxiliary neuron that solves* $\text{WTA}(n, t_c, t_s, \delta)$ *with* $t_s > 10t_c \cdot \frac{\ln(2t_c)}{\ln n}$.

We note that Theorem 5.4.3 applies to simple SNNs (Definition 5.4.1). However, we conjecture that the result holds for any SNN, without structural restrictions. We leave proving a more general bound as an open question, for now.

Proof Outline. We will prove Theorem 5.4.3 assuming that the single auxiliary neuron is an inhibitor. It will be easy to see that a nearly identical proof goes through when the auxiliary neuron is excitatory. At a high level, our proof shows that the convergence and stability inhibitors employed by the two-inhibitor networks described Section 5.3 are necessary. A single auxiliary neuron is not able to both drive fast convergence to a valid WTA output configuration and to maintain stability once the network is in such a state. In more detail, our proof breaks into three steps:

1. We show in Lemma 5.4.4 that the network must be relatively 'active' if it solves $\text{WTA}(n, t_c, t_s, \delta)$. Specifically, if the single inhibitor a does not fire, then any output corresponding to a firing input must fire with probability $\geq 1 - \delta^{1/t_c}$. Otherwise, starting from a configuration in which no outputs fire, the network would take longer than t_c steps to reach a valid WTA output configuration with probability $1 - \delta$.

2. Conversely, we show in Lemma 5.4.8 that if the inhibitor a does fire, then any output with a firing input must *cease firing* at the next time with probability $\geq 1 - n^{-\frac{1}{10t_c}}$. Otherwise, starting from a configuration in which $\Omega(n)$ outputs fire, with probability $\geq 1/2$, the network would take longer than t_c steps to converge to a valid WTA output configuration (with a single firing output).

3. We combine these results a network with just one inhibitor cannot maintain a valid WTA output configuration for $t_s = \Omega\left(t_c \cdot \frac{\ln t_c}{\ln n}\right)$ consecutive steps with probability $\geq 1/2$ (i.e., the network cannot achieve sufficient stability).

   Consider any time t in which $\mathcal{N}$ is in a valid WTA output configuration. If a does not fire at time t, then by (1), if there are $\Omega(n)$ active inputs, at least one output which did not fire at time t fires with probability $\geq 1 - \delta^{\Omega(n/t_c)}$ at time $t+1$. If a does fire, then by (2) the winning output stops firing at time $t+1$ with probability $\geq 1 - n^{-\frac{1}{10t_c}}$. In any case, if $\delta \leq 1/2$, with probability $\geq 1 - n^{-\frac{1}{10t_c}}$, the output configuration changes, and stability is broken.

Since this relatively high probability of breaking stability holds at any time in which $\mathcal{N}$ is in a valid WTA output configuration, it is enough to show that stability cannot be maintained with probability $\geq \epsilon$ for $\Omega\left(t_c \cdot \frac{\ln 1/\epsilon}{\ln n}\right)$ steps. By setting $\epsilon = O(t_c)$ and applying a union bound, we can show that, with probability $\geq 1/2$, in the first t_c time steps, $\mathcal{N}$ never reaches a valid WTA output configuration and remains in this configuration for t_s consecutive steps. Thus, $\mathcal{N}$ does not solve $\mathrm{WTA}(n, t_c, t_s, \delta)$ for $\delta \leq 1/2$.

We start by showing that if the inhibitor a does not fire at time t, then any output corresponding to a firing input must fire with reasonably high probability at time $t+1$. We in fact prove a more general lemma, for networks containing any number of inhibitors, since this result will be useful in our lower bound for two-inhibitor networks presented in Section 5.4.2. The proof is not complicated by adding more inhibitors.

Lemma 5.4.4 (Output Firing Probability When No Inhibitors Are Active)**.** *Let $\mathcal{N} = \langle N, w, b, f\rangle$ be any simple SNN which solves* $\mathrm{WTA}(n, t_c, t_s, \delta)$ *and whose auxiliary neurons $a_1, ..., a_m$ are all inhibitory. For any i, any configuration C with $C(x_i) = 1$ and $C(a_j) = 0$ for all j, and any t,*

$$\mathbb{P}[y_i^{t+1} = 1 | N^t = C] \geq 1 - \delta^{1/t_c}.$$

Proof. Consider any $i \in \{1, ..., n\}$. Assume for the sake of contradiction that there is some configuration C with $C(x_i) = 1$, $C(a_j) = 0$ for all j, and

$$\mathbb{P}[y_i^{t+1} = 1 | N^t = C] < 1 - \delta^{1/t_c}.$$

This assumption additionally implies three claims. First we can see that, since y_i is excitatory,

Claim 5.4.5. *There exists some configuration C with $C(x_i) = 1$, $C(a_j) = 0$ for all j and $C(y_i) = 0$ with $\mathbb{P}[y_i^{t+1} = 1 | N^t = C] < 1 - \delta^{1/t_c}$.*

From Claim 5.4.5 we can additionally conclude that, since, by Definition 5.4.1, y_i can only have connections from itself, x_i, and $a_1, ..., a_m$:

Claim 5.4.6. *For any configuration C with $C(x_i) = 1$, $C(a_j) = 0$ for all j and $C(y_i) = 0$, $\mathbb{P}[y_i^{t+1} = 1 | N^t = C] < 1 - \delta^{1/t_c}$.*

Finally, Claim 5.4.6 implies that, since $a_1, ..., a_m$ are all inhibitors,

Claim 5.4.7. *For any configuration with $C(x_i) = 1$ and $C(y_i) = 0$, $\mathbb{P}[y_i^{t+1} = 1 | N^t = C] < 1 - \delta^{1/t_c}$.*

Consider input execution α_X with X^t fixed for all t (i.e., $X^t = X^{t'}$ for any t, t'), $x_i^t = 1$, and $x_j^t = 0$ for all $j \neq i$. Let N^0 be any initial configuration of $\mathcal{N}$ consistent with α_X (i.e., $N^0(X) = X^0$) and with $y_i^0 = 0$.

Since $\mathcal{N}$ solves WTA(n, t_c, t_s, δ), an infinite output execution drawn from $\mathcal{D}_Y(\mathcal{N}, N^0, \alpha_X)$ must, with probability $\geq 1 - \delta$, reach a valid WTA output configuration (Definition 5.2.6) for some $t \leq t^c$. In particular, with probability $\geq 1 - \delta$, there must be some $t \leq t_c$ in which $y_i^t = 1$. So, letting $\mathcal{E}_0(t)$ be the event that $y_i^{t'} = 0$ for all $t' \leq t$, we have $\mathbb{P}[\mathcal{E}_0(t_c)] \leq \delta$.

Additionally, $\mathbb{P}[\mathcal{E}_0(0)] = 1$ and using Claim 5.4.7 above and inducting on t, for any $t \geq 1$:

$$\begin{aligned}
\mathbb{P}[\mathcal{E}_0(t)] &= \mathbb{P}[\mathcal{E}_0(t) | \mathcal{E}_0(t-1)] \cdot \mathbb{P}[\mathcal{E}_0(t-1)] && \text{(Since } \mathcal{E}_0(t) \subseteq \mathcal{E}_0(t-1).\text{)} \\
&> (1 - (1 - \delta^{1/t_c}))^t = \delta^{t/t_c}.
\end{aligned}$$

We thus have

$$\mathbb{P}[\mathcal{E}_0(t_c)] > \delta^{t_c/t_c} = \delta$$

which contradicts that fact that $\mathbb{P}[\mathcal{E}_0(t_c)] \leq \delta$, giving the lemma.

□

We next show that if the inhibitor a does fire at time t, then any output must cease firing at time $t+1$ with reasonably large probability. Formally, we show this statement for roughly half of the n outputs. It may be possible that some outputs fire with large probability at time $t+1$ whenever their corresponding inputs fire at time t. However, for convergence to occur rapidly give an input execution in which all inputs, this cannot be the case for most outputs.

Again, since it will be useful in the two-inhibitor lower bound proven in Section 5.4.2, we show a more general result which pertains to networks with any number of auxiliary inhibitors.

Lemma 5.4.8 (Output Firing Probability When All Inhibitors Are Active)**.** *Let $\mathcal{N} = \langle N, w, b, f \rangle$ be any simple SNN which solves* WTA(n, t_c, t_s, δ) *for $n \geq 20$ and $\delta \leq 1/2$ and whose auxiliary neurons $a_1, ..., a_m$ are all inhibitory. There is some set $\mathcal{S} \subseteq \{1, ..., n\}$ with $|\mathcal{S}| \geq \lceil n/2 \rceil$ such that, for any $i \in \mathcal{S}$, any configuration C with $C(x_i) = 1$ and $C(a_j) = 1$ for all j, and any t,*

$$\mathbb{P}[y_i^{t+1} = 0 | N^t = C] \geq 1 - n^{-\frac{1}{10t_c}}.$$

Proof. Assume for the sake of contradiction that there is some set $\mathcal{R} \subseteq \{1, ..., n\}$ with $|\mathcal{R}| = \lfloor n/2 \rfloor + 1$ such that, for each $i \in \mathcal{R}$, there exists some configuration C with $C(x_i) = 1$, $C(a_j) = 1$ for all j and

$$\mathbb{P}[y_i^{t+1} = 1 | N^t = C] > n^{-\frac{1}{10t_c}}.$$

From this assumption we can deduce three claims. First, since each y_i is excitatory,

Claim 5.4.9. *For each $i \in \mathcal{R}$ there exists some configuration C with $C(x_i) = 1$, $C(a_j) = 1$ for all j and $C(y_i) = 1$ with $\mathbb{P}[y_i^{t+1} = 1 | N^t = C] > n^{-\frac{1}{10t_c}}$.*

Claim 5.4.9 further implies, since y_i can only have connections from itself, x_i, and $a_1, ..., a_m$ (see Definition 5.4.1):

Claim 5.4.10. *For any configuration C with $C(x_i) = 1$, $C(a_j) = 1$ for all j and $C(y_i) = 1$, $\mathbb{P}[y_i^{t+1} = 1 | N^t = C] > n^{-\frac{1}{10t_c}}$.*

Finally, from Claim 5.4.10 and the fact that $a_1, ..., a_m$ are all inhibitors we can conclude:

Claim 5.4.11. *For any configuration with $C(x_i) = 1$ and $C(y_i) = 1$, $\mathbb{P}[y_i^{t+1} = 1 | N^t = C] > n^{-\frac{1}{10t_c}}$.*

Let α_X be any infinite input execution with X^t fixed for all t and $X^t(x_i) = 1$ for all i. Let N^0 be any initial configuration of $\mathcal{N}$ consistent with α_X (i.e., $N^0(X) = X^0$) and with $Y^0(y_i) = 1$ for all i.

Since $\mathcal{N}$ solves WTA(n, t_c, t_s, δ), an infinite output execution drawn from $\mathcal{D}_Y(\mathcal{N}, N^0, \alpha_X)$ must, with probability $\geq 1 - \delta$, reach a valid WTA output configuration (Definition 5.2.6) for some $t \leq t^c$. In particular, with probability $\geq 1 - \delta$, there must be some $t \leq t_c$ in which $y_i^t = 0$ for at most one $i \in \mathcal{R}$.

Let $\mathcal{I}_{act}(t, i) \in \{0, 1\}$ be an indicator of the event that $y_i^{t'} = 1$ for all $t' \leq t$. If $\mathcal{I}_{act}(t_c, i) = 1$ for more than one $i \in \mathcal{R}$, then the network has not reached a valid

WTA output state within time t_c. Thus, since $\mathcal{N}$ solves WTA(n, t_c, t_s, δ) we have:

$$\mathbb{P}\left[\sum_{i \in \mathcal{R}} \mathcal{I}_{act}(t_c, i) \geq 2\right] \leq \delta. \tag{5.38}$$

We can bound $\mathbb{P}\left[\sum_{i \in \mathcal{R}} \mathcal{I}_{act}(t_c, i) \geq 2\right]$ from below using Claim 5.4.11 above and Lemma 5.2.4. Define for all i and t a random variable $Z_{i,t} \in \{0, 1\}$ which is set to 1 independently with probability $n^{-\frac{1}{10t_c}}$. Let $\bar{\mathcal{I}}(t_c, i)$ be an indicator of the event that $Z_{i,1} = ... = Z_{i,t_c} = 1$. Clearly the $\bar{\mathcal{I}}(t_c, i)$ variables are independent and $\mathbb{P}[\bar{\mathcal{I}}(t_c, i) = 1] = \left(n^{-\frac{1}{10t_c}}\right)^{t_c} = n^{-\frac{1}{10}}$.

By (5.38) and Lemma 5.2.4 we thus have:

$$\begin{aligned}
\delta \geq \mathbb{P}\left[\sum_{i \in \mathcal{R}} \mathcal{I}_{act}(t_c, i) \geq 2\right] &\geq \mathbb{P}\left[\sum_{i \in \mathcal{R}} \bar{\mathcal{I}}(t_c, i) \geq 2\right] \\
&= 1 - \left(1 - n^{-\frac{1}{10}}\right)^{|\mathcal{R}|} - |\mathcal{R}| \left(1 - n^{-\frac{1}{10}}\right)^{|\mathcal{R}|-1} \\
&\geq 1 - n\left(1 - n^{-\frac{1}{10}}\right)^{n/4}
\end{aligned}$$

where in the last step we bound $|\mathcal{R}| - 1 \geq \lfloor n/2 \rfloor \geq n/4$. Rearranging:

$$\frac{1 - \delta}{n} \leq \left(1 - n^{-\frac{1}{10}}\right)^{n/4} \leq e^{-\frac{n^{9/10}}{4}}. \tag{5.39}$$

We can check that whenever $n \geq 20$, $e^{-\frac{n^{9/10}}{4}} < \frac{1}{2n} < \frac{1-\delta}{n}$, by our assumption that $\delta \leq 1/2$. This contradicts (5.39), thus giving the lemma. □

We conclude by combining Lemmas 5.4.4 and 5.4.8 to prove Theorem 5.4.3. We first show a simple auxiliary lemma which lower bounds the probability that a valid WTA output configuration at time t remains fixed at time $t+1$ in terms of the network convergence time t_c and stability time t_s. The smaller t_c and the larger t_s, the larger the lower bound on this probability is.

Note that the theorem only lower bounds the probability that the output Y^t stays fixed for *some configuration* C where $C(Y)$ is a valid WTA output. It does not bound this probability for all such configurations, and in fact such a bound cannot be shown for all such configurations. Recall, for example, that for our two-inhibitor networks in Section 5.3, both near-valid WTA configurations (Definition 5.3.15) and valid WTA configurations (Definition 5.3.11) have valid WTA output configurations. However, the output of a near-valid configuration only remains fixed with constant probability (Lemma 5.3.21).

Lemma 5.4.12 (Single Step Stability Probability). *If $\mathcal{N} = \langle N, w, b, f \rangle$ is a simple SNN which solves $WTA(n, t_c, t_s, \delta)$ for $\delta \leq 1/2$, then there must exist some configuration C with $C(Y)$ a valid WTA output configuration such that:*

$$\mathbb{P}[Y^{t+1} = Y^t | N^t = C] \geq \frac{1}{(2t_c)^{1/t_s}} \geq 1 - \frac{\ln 2t_c}{t_s}. \tag{5.40}$$

Proof. The second inequality simply follows since $e^{-x} \geq 1 - x$ for all x. So we focus on proving the first inequality. Assume for the sake of contradiction that for every configuration C with $C(Y)$ a valid WTA output configuration,

$$\mathbb{P}[Y^{t+1} = Y^t | N^t = C] < \frac{1}{(2t_c)^{1/t_s}}.$$

For any t, i, let $\mathcal{E}_{stab}(t, i)$ be the event that $Y^t = Y^{t+1} = ... = Y^{t+i}$. Trivially $\mathbb{P}[\mathcal{E}_{stab}(t, 0)] = 1$ for all t. Using the assumption of (5.40) and induction we can bound for any $i \geq 1$:

$$\begin{aligned}
\mathbb{P}[\mathcal{E}_{stab}(t,i)|N^t = C] &= \mathbb{P}[\mathcal{E}_{stab}(t,i)|\mathcal{E}_{stab}(t,i-1), N^t = C] \cdot \mathbb{P}[\mathcal{E}_{stab}(t,i-1)|N^t = C] \\
&\qquad \text{(Since } \mathcal{E}_{stab}(t,i-1) \subseteq \mathcal{E}_{stab}(t,i)\text{.)} \\
&< \frac{1}{(2t_c)^{1/t_s}} \cdot \mathbb{P}[\mathcal{E}_{stab}(t,i-1)|N^t = C] \\
&\leq \left(\frac{1}{(2t_c)^{1/t_s}}\right)^i.
\end{aligned}$$

Thus, for any C with $C(Y)$ a valid WTA output configuration,

$$\mathbb{P}[\mathcal{E}_{stab}(t, t_s)|N^t = C] < \left(\frac{1}{(2t_c)^{1/t_s}}\right)^{t_s} < \frac{1}{2t_c}. \tag{5.41}$$

Let $\mathcal{E}$ be the event that $\mathcal{N}$ reaches a valid WTA output configuration within $t \leq t_c$ steps and remains in this output configuration for t_s consecutive steps. Let $Z_1, ..., Z_{t_c}$ be i.i.d. random variables with $Z_t = 1$ with probability $\frac{1}{2t_c}$ and $Z_t = 0$ otherwise.

Invoking (5.41) and Lemma 5.2.3,

$$\begin{aligned}\mathbb{P}[\mathcal{E}] < \mathbb{P}\left[\sum_{t=1}^{t_c} Z_t \geq 1\right] &= 1 - \left(1 - \frac{1}{2t_c}\right)^{t_c}\\ &< 1/2\\ &< 1-\delta \qquad (5.42)\end{aligned}$$

for $\delta \leq 1/2$. This contradicts the fact that $\mathcal{N}$ solves WTA(n, t_c, t_s, δ), giving the lemma. □

We can now combine Lemmas 5.4.4 and 5.4.8 to show that in a single-inhibitor network, for any configuration C with $C(Y)$ a valid WTA output configuration $\mathbb{P}[Y^{t+1} = Y^t | N^t = C]$ cannot be too large. This contradicts Lemma 5.4.12, giving our lower bound.

Proof of Theorem 5.4.3.

Consider any simple SNN $\mathcal{N} = \langle N, w, b, f\rangle$ with just a single auxiliary neuron a, which solves WTA(n, t_c, t_s, δ) for $n \geq 20$, $\delta \leq 1/2$, $t_s > 10t_c \cdot \frac{\ln(2t_c)}{\ln n}$. Let $\mathcal{S}$ be the set of indices shown to exist in Lemma 5.4.8, which do not fire with too high probability when all inhibitors in the network fire.

Let α_X be the infinite input execution with X^t fixed for all t, $X^t(x_i) = 1$ for all $i \in \mathcal{S}$ and $X^t(x_i) = 0$ for all $i \notin \mathcal{S}$. Let N^0 be any initial configuration of $\mathcal{N}$ consistent with α_X.

Since $\mathcal{N}$ solves WTA(n, t_c, t_s, δ), an infinite output execution drawn from $\mathcal{D}_Y(\mathcal{N}, N^0, \alpha_X)$ must, with probability $\geq 1-\delta$, reach a valid WTA output state (Definition 5.2.6) for some $t \leq t_c$ and remain in this state for t_s consecutive steps. Let C be any configuration where $C(Y)$ is a valid WTA output configuration for α_X. We must have for exactly one $i \in \mathcal{S}$, $C(y_i) = 1$. Let $\mathcal{E}_{fail}(t)$ be the event that $Y^{t+1} \neq Y^t$. We consider two cases:

Case 1: $C(a) = 0$.

In this case, by Lemma 5.4.4, for all $j \in \mathcal{S}$, y_j fires with probability $\geq 1 - \delta^{1/t_c}$ at time $t+1$. So, with probability $\geq 1 - \delta^{\frac{|\mathcal{S}|-1}{t_c}} \geq 1 - \delta^{\frac{n}{4t_c}}$ as least one output other than the winner fires at time $t+1$. This gives:

$$\mathbb{P}[\mathcal{E}_{fail}(t) | N^t = C] \geq 1 - \delta^{\frac{n}{4t_c}} \geq 1 - n^{-\frac{1}{4t_c}}, \qquad (5.43)$$

where the second inequality follows from the assumption that $n \geq 20$ and $\delta \leq 1/2$.

Case 2: $C(a) = 1$.

In this case, Lemma 5.4.8 gives that, conditioned on $N^t = C$, y_i does not fire at time $t+1$ with probability $\geq 1 - n^{-\frac{1}{10t_c}}$. This gives:

$$\mathbb{P}[\mathcal{E}_{fail}(t) | N^t = C] \geq 1 - n^{-\frac{1}{10t_c}}. \tag{5.44}$$

So overall, combining (5.43) and (5.44) we have for C with $C(Y)$ a valid WTA output configuration,

$$\mathbb{P}[\mathcal{E}_{fail}(t) | N^t = C] \geq 1 - n^{-\frac{1}{10t_c}}$$

and so

$$\mathbb{P}[Y^{t+1} = Y^t | N^t = C] \leq n^{-\frac{1}{10t_c}}.$$

For $t_s > 10t_c \cdot \frac{\ln(2t_c)}{\ln n}$ this gives

$$\mathbb{P}[Y^{t+1} = Y^t | N^t = C] < n^{-\frac{\ln 2t_c}{t_s \cdot \ln n}} < \frac{1}{(2t_c)^{1/t_s}}. \tag{5.45}$$

This contradicts Lemma 5.4.12, giving the theorem. □

Remark on the Tightness of Theorem 5.4.3. We note that our proof of Theorem 5.4.3 is loose. In bounding $\mathbb{P}[\mathcal{E}]$ in (5.42), we do not consider the time required to converge to a valid WTA output state, or the time spent in this converged state before convergence is broken. We conjecture that if this time were taken into account, it would be possible improve the $\ln(2t_c)$ term, as well as add a dependence on the failure probability δ, giving a lower bound of $t_s = O\left(\frac{t_c}{\log n \cdot \log(1/\delta)}\right)$.

We note that by simply removing the stability inhibitor from our two-inhibitor network family presented in Section 5.3, we obtain a family of single inhibitor WTA networks with $t_s = 1$ and $t_c = O(\log n \cdot \log(1/\delta))$. This matches the conjectured stronger lower bound above up to a constant factor since for $t_c = O(\log n \cdot \log(1/\delta))$ we have $O\left(\frac{t_c}{\log n \cdot \log(1/\delta)}\right) = O(1)$.

Theorem 5.4.13 (Single-Inhibitor WTA Network)**.** *There exists a simple SNN with a single inhibitory auxiliary neuron which solves $WTA(n, t_c, t_s, \delta)$ for $t_s = 1$ and $t_c = O(\log n \cdot \log(1/\delta))$.*

Proof Sketch. Let $\mathcal{T}'_{n,\gamma}$ be identical to $\mathcal{T}_{n,\gamma}$ as described in Section 5.3.1, but with a_s removed from the network and with $w(a_c, y_i) = -2\gamma$ for all i. In $\mathcal{T}_{n,\gamma}$, $w(a_c, y_i) = w(a_s, y_i) = -\gamma$ so $w(a_c, y_i) + w(a_s, y_i) = -2\gamma$. Thus, in $\mathcal{T}'_{n,\gamma}$, when $a_c^t = 1$, the firing probabilities of all neurons at time $t+1$ are identical to what they would be in $\mathcal{T}_{n,\gamma}$ assuming the same configuration at time t but with $a_c^t = a_s^t = 1$.

Using this equivalence, it is tedious but easy to check that an analogous result to Theorem 5.3.32 holds, where a valid WTA configuration is redefined to be any configuration C where $C(Y)$ is a valid WTA output configuration and $C(a_c) = 1$. By a similar to result to Lemma 5.3.21, with probability $1/2 - ne^{-\gamma/2}$, if $\mathcal{T}'_{n,\gamma}$ is in such a configuration at time t, it is also in a valid WTA output configuration at time $t+1$. Thus, with $\Theta(1)$ probability, a valid WTA output configuration is reached and maintained for $t_s = 1$ steps within $O(\log n)$ steps. We can then use an argument similar to that of Theorem 5.3.2 to argue that in $O(\log n \cdot \log(1/\delta))$ steps a valid WTA output configuration is reached and maintained for 1 step with probability $\geq 1 - \delta$. □

5.4.2 Two Auxiliary Neuron Lower Bound

We next give a convergence time lower bound for SNNs with two auxiliary neurons, showing that the rate given by the family of two-inhibitor networks $\mathcal{T}_{n,\gamma}$ presented in Section 5.3 is optimal up to a $O(\log \log n)$ factor. To simplify our argument, we focus the further restricted class of symmetric SNNs described in Definition 5.4.2, proving:

Theorem 5.4.14 (Two Neuron Lower Bound)**.** *For any $n \geq 341$, $\delta \leq 1/2$, and any spike probability function $f : \mathbb{R} \to [0,1]$ (satisfying the restrictions in Section 5.2.1), there is no symmetric SNN $\mathcal{N} = \langle N, w, b, f \rangle$ using two auxiliary neurons that solves* $\mathrm{WTA}(n, t_c, t_s, \delta)$ *with $t_c \leq \frac{\ln n}{30 \ln \ln n}$ and $t_s \geq 32 \ln n \cdot \ln 2t_c$.*

As with Theorem 5.4.3, we conjecture that this result holds more generally, for any SNN with two auxiliary neurons (i.e., without making the assumptions of Definition 5.4.2). In the theorem we require $t_s \geq 32 \ln n \cdot \ln 2t_c$. However, we conjecture that the result holds even for $t_s = O(1)$.

Proof Outline. Our proof of Theorem 5.4.14 is similar in spirit to that of Theorem 5.4.3. We consider the case when both auxiliary neurons are inhibitors and note that a similar proof applies when one or both of the neurons are excitatory.

With two inhibitors we have to not only consider the cases when neither inhibitor fires (analyzed in Lemma 5.4.4) and when both inhibitors fire (analyzed in Lemma

5.4.8), but also the cases in which one of the inhibitors fires. Our analysis breaks down as follows:

1. By Lemma 5.4.4, if neither inhibitor fires at time t, then at time $t+1$ any output corresponding to a firing input must fire with probability $\geq 1 - \delta^{1/t_c} = \Omega(\frac{1}{\ln n})$ when $\delta \leq 1/2$ and $t_c \leq \frac{\ln n}{30 \ln \ln n}$. By Lemma 5.4.8, if both inhibitors fire at time t, then at time $t+1$, any output with a firing input must *not fire* at time $t+1$ with probability $\geq 1 - n^{\frac{1}{10t_c}} \geq 1 - \frac{1}{\ln^3 n}$ when $t_c \leq \frac{\ln n}{30 \ln \ln n}$ and n is sufficiently large.

2. In Lemma 5.4.16 we show that, by the above bounds, if we consider an input with $\Theta(\ln^2 n)$ firing inputs, then if either both inhibitors fire or neither inhibitor fires at time t, except with probability $\leq \frac{1}{\ln n}$, at time $t+1$, either 0 or ≥ 2 outputs will fire, and so $\mathcal{N}$ will not be in a valid WTA output configuration.

3. In Lemma 5.4.18 we show that, in order for $\mathcal{N}$ to stabilize to a valid WTA output configuration for t_s steps, if Y^t is a valid WTA output configuration, then *exactly one inhibitor* must fire with good probability at time $t+1$. Otherwise, if neither or both inhibitors fire at time $t+1$, by Lemma 5.4.16, Y^{t+2} is unlikely to be a valid WTA output configuration. So if t_s is large, over t_s steps, stability is likely to be broken at some point.

4. In Lemma 5.4.20 we prove that, since the inhibitors fire independently at time $t+1$ conditioned on N^t, Lemma 5.4.18 in fact requires that one of the inhibitors fires with high probability at time $t+1$ when Y^t is a valid WTA output configuration and that the other inhibitor is silent with high probability.

   By our symmetry assumption (Definition 5.4.2), given a fixed input firing pattern, the probability that an inhibitor fires at time $t+1$ depends only on the number of outputs that fire at time t. So we can in fact show a stronger result: one inhibitor (assume without loss of generality a_1) fires with high probability at time $t+1$ when $\|Y^t\|_1 = 1$ while the other (assume without loss of generality a_2) remains silent with high probability at time $t+1$ when $\|Y^t\|_1 = 1$. Since all outputs are excitatory, this implies that a_1 fires with high probability whenever $\|Y^t\|_1 \geq 1$, while a_2 does not fire with high probability whenever $\|Y^t\|_1 \leq 1$.

   This result shows that, in any two-inhibitor network with fast convergence and a reasonably long stability period t_s, the inhibitors must exhibit separate behaviors. Neuron a_1 fires with high probability at time $t+1$ whenever one output fires at time t, maintaining stability of valid WTA configurations. Neuron a_2 is silent with high probability at time $t+1$, except possibly when $\|Y^t\|_1 \geq 2$. We can see this separation, for example, in the behavior of a_s and a_c in the two-inhibitor $\mathcal{T}_{n,\gamma}$ networks analyzed in Section 5.3 (see Lemma 5.3.7). The stability inhibitor, a_s, fires with high probability at time $t+1$ when at least one output fires at time t (e.g., when the network is in a valid WTA configuration). The convergence inhibitor, a_c, only fires with high probability at time $t+1$ when at least two outputs fire at time t.

5. In Corollary 5.4.25 we use Lemma 5.4.18 to show that it is unlikely that a_2 ever fires at a time in which a_1 does not, which will be useful in our eventual case analysis (see Step 7).

6. In Lemma 5.4.26 we show that, since when a_1 fires alone it must maintain stability of a valid WTA output configuration, when a_1 fires at time t, any output with an active input which fired at time t is likely to continue firing at time $t+1$. Any output that did not fire at time t is unlikely to fire at time $t+1$. This result shows that a_1 *must act as a stability inhibitor*, reflecting the role of a_s in the two-inhibitor network family presented in Section 5.3.

   In Corollary 5.4.27 we show that Lemma 5.4.26 implies that, if Y^t is *not a valid WTA output configuration* and just a_1 fires at time t, since the output firing states are maintained with high probability at time $t+1$, Y^{t+1} is unlikely to be a valid WTA output configuration.

7. We finally prove Theorem 5.4.14 via a case analysis. By Lemma 5.4.16, if 0 or 2 inhibitors fire at time t, Y^{t+1} is unlikely to be a valid WTA output configuration (see Step 2 above). By Corollary 5.4.27, if $a_1^t = 1$, $a_2^t = 0$, and Y^t is not a valid WTA output configuration, then Y^{t+1} is unlikely to be a valid WTA output configuration (see Step 6 above). Finally, it is unlikely that we ever have $a_1^t = 0$, $a_2^t = 1$ (see Step 5 above). Thus, convergence to a valid WTA output configuration is relatively unlikely at all times, letting us prove a lower bound on convergence time.

We first define a *hard input execution*, based on Lemma 5.4.8, which shows that at least half of the output neurons must not fire with too high probability when all inhibitors in the network fire. Specifically, Lemma 5.4.8 guarantees the existence of some set $\mathcal{S}$ with $|\mathcal{S}| \geq \lceil n/2 \rceil$ such that for any $i \in \mathcal{S}$, any configuration C of $\mathcal{N}$ with

$C(x_i) = 1$ and $C(a_1) = C(a_2) = 1$, and any time t,

$$\mathbb{P}[y_i^{t+1} = 0 | N^t = C] \geq 1 - n^{-\frac{1}{10t_c}}.$$

Since in Theorem 5.4.14 we require $n \geq 341$ we easily have $\lceil n/2 \rceil \geq \lceil \ln^2 n \rceil$ and so can construct a hard input execution as follows:

Definition 5.4.15 (Hard Input Execution). *Let $\mathcal{N} = \langle N, w, b, f \rangle$ be any simple SNN with two auxiliary inhibitory neurons a_1, a_2 which solves* WTA(n, t_c, t_s, δ) *for $n \geq 341$ and $\delta \leq 1/2$. Fix any set $\mathcal{R} \subseteq \{1, ..., n\}$ with $|\mathcal{R}| = \lfloor \ln^2 n \rfloor$ such that for any $i \in \mathcal{R}$, any C with $C(x_i) = 1$ and $C(a_1) = C(a_2) = 1$, and any time t,*

$$\mathbb{P}[y_i^{t+1} = 0 | N^t = C] \geq 1 - n^{-\frac{1}{10t_c}}.$$

Let $X_{hard,\mathcal{N}}$ be the input configuration with $X_{hard,\mathcal{N}}(x_i) = 1$ for all $i \in \mathcal{R}$, and $X_{hard,\mathcal{N}}(x_i) = 0$ for all $i \notin \mathcal{R}$. Let $\alpha_{hard,\mathcal{N}}$ be the infinite input execution with $X^t = X_{hard,\mathcal{N}}$ for all t.

Lemma 5.4.16 (Valid WTA is Unlikely After Both or Neither Inhibitors Fire). *Let $\mathcal{N} = \langle N, w, b, f \rangle$ be any simple SNN with two auxiliary inhibitory neurons a_1, a_2 which solves $WTA(n, t_c, t_s, \delta)$ for $n \geq 341$, $\delta \leq 1/2$, and $t_c \leq \frac{\ln n}{30 \ln \ln n}$. For any configuration C with $C(X) = X_{hard,\mathcal{N}}$ (Definition 5.4.15) and with $C(a_1) = C(a_2) = 1$ or $C(a_1) = C(a_2) = 0$,*

$$\mathbb{P}[\|Y^{t+1}\| = 1 | N^t = C] \leq \frac{1}{\ln n}.$$

Proof. We prove the lemma in two cases depending on the inhibitor behavior.

Case 1: $C(a_1) = C(a_2) = 0$**.**

In this case, by Lemma 5.4.4, since $X^t(x_i) = X_{hard,\mathcal{N}}(x_i) = 1$ for any $i \in \mathcal{R}$ by Definition 5.4.15,

$$\mathbb{P}[y_i^{t+1} = 1 | N^t = C] \geq 1 - \delta^{1/t_c} \geq 1 - \frac{1}{2^{\frac{\ln \ln n}{\ln n}}}$$

where the second inequality follows from our assumption that $\delta \leq 1/2$ and $t_c \leq \frac{\ln n}{30 \ln \ln n} \leq \frac{\ln n}{\ln \ln n}$. For any $x \in [1, 2]$, $(1 - 1/x) \geq \frac{\log_2 x}{2}$. Since $2^{\frac{\ln \ln n}{\ln n}} \in [1, 2]$ this gives:

$$\mathbb{P}[y_i^{t+1} = 1 | N^t = C] \geq \frac{\ln \ln n}{2 \ln n}. \tag{5.46}$$

Using (5.46) we can bound the probability that $\|Y^{t+1}\|_1 \neq 1$ by:

$$\mathbb{P}[\|Y^{t+1}\|_1 \neq 1 | N^t = C] \geq 1 - |\mathcal{R}| \left(1 - \frac{\ln \ln n}{2 \ln n}\right)^{|\mathcal{R}|-1}.$$

We can check numerically that since $|\mathcal{R}| = \lfloor \ln^2 n \rfloor$ and by assumption $n \geq 341$, the above can be lower bounded to give:

$$\mathbb{P}[\|Y^{t+1}\|_1 \neq 1 | N^t = C] \geq 1 - \frac{1}{\ln n}.$$

This gives $\mathbb{P}[\|Y^{t+1}\|_1 = 1 | N^t = C] \leq \frac{1}{\ln n}$ and thus the lemma in this case.

Case 2: $C(a_1) = C(a_2) = 1$.

In this case, by Lemma 5.4.8 and the assumption that $t_c \leq \frac{\ln n}{30 \ln \ln n}$, for any $i \in \mathcal{R}$:

$$\mathbb{P}[y_i^{t+1} = 1 | N^t = C] \leq n^{-\frac{1}{10 t_c}} \leq e^{-\frac{30 \ln \ln n}{10}} \leq \frac{1}{\ln^3 n}. \tag{5.47}$$

Using (5.47) we can bound:

$$\begin{aligned} \mathbb{P}[\|Y^{t+1}\|_1 = 1 | N^t = C] &\leq \frac{|\mathcal{R}|}{\ln^3 n} \\ &= \frac{\lfloor \ln^2 n \rfloor}{\ln^3 n} \\ &\leq \frac{1}{\ln n}, \end{aligned}$$

which gives the lemma in this case.

□

By Lemma 5.4.16, if $a_1^t = a_2^t = 0$ or $a_1^t = a_2^t = 1$, then Y^{t+1} is unlikely to have $\|Y^{t+1}\|_1 = 1$ and thus is unlikely to be a valid WTA output configuration. Thus, to stabilize to any valid WTA output configuration C, if $N^t = C$ exactly one of a_1 or a_2 must fire at time $t+1$. Otherwise convergence will likely be broken at time $t+2$. To prove this, we first show an auxiliary lemma, very similar to Lemma 5.4.12, which bounds the probability of maintaining stability over two time steps in terms of the convergence and stability times t_c and t_s.

Lemma 5.4.17 (Lower Bound on the Two-Step Stability Probability). *If $\mathcal{N} = \langle N, w, b, f \rangle$ is a simple SNN which solves $WTA(n, t_c, t_s, \delta)$ for $\delta \leq 1/2$, then there* *exists some configuration C with $C(Y)$ a valid WTA output configuration such that:*

$$\mathbb{P}[Y^{t+2} = Y^{t+1} = Y^t | N^t = C] \geq 1 - \frac{4 \ln 2t_c}{t_s}. \tag{5.48}$$

Proof. Our proof mirrors that of Lemma 5.4.12. Assume for the sake of contradiction that for any C with $C(Y)$ a valid WTA output configuration,

$$\mathbb{P}[Y^{t+2} = Y^{t+1} = Y^t | N^t = C] < 1 - \frac{4 \ln 2t_c}{t_s}.$$

For any t, i, let $\mathcal{E}_{stab}(t,i)$ be the event that $Y^t = Y^{t+1} = ... = Y^{t+i}$. Trivially, $\mathbb{P}[\mathcal{E}_{stab}(t,0)] = 1$ for all t. Using the assumption if (5.48) and induction we can bound for any $i \geq 2$:

$$\begin{aligned}
\mathbb{P}[\mathcal{E}_{stab}(t,i)|N^t = C] &= \mathbb{P}[\mathcal{E}_{stab}(t,i)|\mathcal{E}_{stab}(t,t-2), N^t = C] \cdot \mathbb{P}[\mathcal{E}_{stab}(t,i-2)|N^t = C] \\
&\qquad\qquad\qquad (\text{Since } \mathcal{E}_{stab}(t,i-2) \subseteq \mathcal{E}_{stab}(t,i).) \\
&< \left(1 - \frac{4 \ln 2t_c}{t_s}\right) \cdot \mathbb{P}[\mathcal{E}_{stab}(t,i-2)|N^t = C] \\
&\leq \left(1 - \frac{4 \ln 2t_c}{t_s}\right)^{\lfloor i/2 \rfloor}.
\end{aligned}$$

Thus, for any C with $C(Y)$ a valid WTA output configuration,

$$\begin{aligned}
\mathbb{P}[\mathcal{E}_{stab}(t,t_s)|N^t = C] &< \left(1 - \frac{4 \cdot \ln 2t_c}{t_s}\right)^{\lfloor t_s/2 \rfloor} \\
&< \left(1 - \frac{4 \cdot \ln 2t_c}{t_s}\right)^{t_s/4} \\
&< \frac{1}{2t_c}.
\end{aligned} \tag{5.49}$$

Let $\mathcal{E}$ be the event that $\mathcal{N}$ reaches a valid WTA output configuration within $t \leq t_c$ steps and remains in this output configuration for t_s consecutive steps. Let $Z_1, ..., Z_{t_c}$ be i.i.d. random variables with $Z_t = 1$ with probability $\frac{1}{2t_c}$ and $Z_t = 0$ otherwise.

Invoking (5.49) and Lemma 5.2.3,

$$\begin{aligned}\mathbb{P}[\mathcal{E}] < \mathbb{P}\left[\sum_{t=1}^{t_c} Z_t \geq 1\right] &= 1 - \left(1 - \frac{1}{2t_c}\right)^{t_c} \\ &< 1/2 \\ &< 1 - \delta \qquad (5.50)\end{aligned}$$

for $\delta \leq 1/2$. This contradicts the fact that $\mathcal{N}$ solves WTA(n, t_c, t_s, δ), giving the lemma. □

We can now use Lemmas 5.4.16 and 5.4.17 to show that, for $\mathcal{N}$ to stabilize to a valid WTA output configuration for t_s steps with good probability, if Y^t is a valid WTA output configuration, then, with high probability, exactly one inhibitor must fire at time $t+1$.

Lemma 5.4.18 (A Single Inhibitor is Likely to Fire After a Valid WTA Configuration)**.** *Let $\mathcal{N} = \langle N, w, b, f\rangle$ be any simple SNN with two auxiliary inhibitory neurons $A = \{a_1, a_2\}$ which solves $WTA(n, t_c, t_s, \delta)$ for $n \geq 341$, $\delta \leq 1/2$, and $t_c \leq \frac{\ln n}{30 \ln \ln n}$. For any configuration C with $C(X) = X_{hard,\mathcal{N}}$ (Definition 5.4.15) and with $C(Y)$ a valid WTA output configuration,*

$$\mathbb{P}[\|A^{t+1}\|_1 = 1 | N^t = C] \geq 1 - \frac{8 \cdot \ln 2t_c}{t_s}.$$

Proof. Assume for the sake of contradiction that there exists C with $C(Y)$ a valid WTA output configuration and

$$\mathbb{P}[\|A^{t+1}\|_1 = 1 | N^t = C] < 1 - \frac{8 \cdot \ln 2t_c}{t_s}.$$

From this assumption, since in a symmetric SNN (Definition 5.4.2), $w(y_i, u) = w(y_j, u)$ for all i, j and $u \in A$, and since each auxiliary neuron may only have connections from the inputs and outputs (not to the other auxiliary neurons), we can deduce the stronger claim:

Claim 5.4.19. *For* every *C with $C(Y)$ a valid WTA output configuration,*

$$\mathbb{P}[\|A^{t+1}\|_1 = 1 | N^t = C] < 1 - \frac{8 \cdot \ln 2t_c}{t_s}.$$

Using Claim 5.4.19 we can bound the probability that $\mathcal{N}$ remains in a valid WTA output configuration at time $t+2$ if it is in a valid configuration at time t. Specifically, for any C with $C(Y)$ a valid WTA output configuration:

$$\begin{aligned}\mathbb{P}[Y^{t+2} = Y^t | N^t = C] &= \mathbb{P}[Y^{t+2} = Y^t | \|A^{t+1}\|_1 = 1, N^t = C] \cdot \mathbb{P}[\|A^{t+1}\|_1 = 1 | N^t = C] \\ &\quad + \mathbb{P}[Y^{t+2} = Y^t | \|A^{t+1}\|_1 \neq 1, N^t = C] \cdot \mathbb{P}[\|A^{t+1}\|_1 \neq 1 | N^t = C] \\ &< \left(1 - \frac{8 \cdot \ln 2t_c}{t_s}\right) + \left(\frac{8 \cdot \ln 2t_c}{t_s}\right) \cdot \mathbb{P}[Y^{t+2} = Y^t | \|A^{t+1}\|_1 \neq 1, N^t = C]. \end{aligned} \tag{5.51}$$

Since $C(Y)$ is a valid WTA output configuration, having $Y^{t+2} = Y^t$ requires that $\|Y^{t+2}\|_1 = 1$. Thus by Lemma 5.4.16 we can bound:

$$\mathbb{P}[Y^{t+2} = Y^t | \|A^{t+1}\|_1 \neq 1, N^t = C] \leq \frac{1}{\ln n}.$$

Plugging back into (5.51), for any C with $C(Y)$ a valid WTA output configuration:

$$\begin{aligned}\mathbb{P}[Y^{t+2} = Y^t | N^t = C] &< \left(1 - \frac{8 \cdot \ln 2t_c}{t_s}\right) + \left(\frac{8 \cdot \ln 2t_c}{t_s}\right) \cdot \left(\frac{1}{\ln n}\right) \\ &< 1 - \frac{4 \cdot \ln 2t_c}{t_s} \end{aligned} \tag{5.52}$$

where the last inequality holds easily since we require $n \geq 341$ and so $\ln n \geq 2$. Using (5.52) we can bound:

$$\mathbb{P}[Y^{t+2} = Y^{t+1} = Y^t | N^t = C] \leq \mathbb{P}[Y^{t+2} = Y^t | N^t = C] < 1 - \frac{4 \cdot \ln 2t_c}{t_s}.$$

This contradicts Lemma 5.4.17, giving the lemma. □

Since at time $t + 1$, conditioned on the configuration at time t, a_1 and a_2 fire independently, we can in fact strengthen Lemma 5.4.18 to show that one of a_1, a_2 fires with high probability at time $t + 1$ when $\|Y^t\|_1 \geq 1$ and that the other remains silent with high probability when $\|Y^t\|_1 \leq 1$.

Lemma 5.4.20 (Separation of Inhibitor Behaviors)**.** *Let $\mathcal{N} = \langle N, w, b, f\rangle$ be any symmetric SNN with two auxiliary inhibitory neurons $A = \{a_1, a_2\}$ which solves $WTA(n, t_c, t_s, \delta)$ for $n \geq 341$, $\delta \leq 1/2$, $t_c \leq \frac{\ln n}{30 \ln \ln n}$, and $\frac{8 \ln 2t_c}{t_s} \leq \frac{1}{4}$. There exists some $i \in \{1, 2\}$ such that, for any configuration C with $C(X) = X_{hard,\mathcal{N}}$ (Definition 5.4.15),*

1. *If* $\|C(Y)\|_1 \geq 1$*, then*

$$\mathbb{P}[a_i^{t+1} = 1 | N^t = C] \geq 1 - \frac{8 \cdot \ln 2t_c}{t_s}.$$

2. *If* $\|C(Y)\|_1 \leq 1$*, then for* $j \neq i$*,*

$$\mathbb{P}[a_j^{t+1} = 1 | N^t = C] \leq \frac{12 \cdot \ln 2t_c}{t_s}.$$

We can assume without loss of generality that $i = 1$*.*

Proof. We prove the two conclusions in sequence. We first show that there exists some i such that conclusion (1) holds. Fixing i, we then show that for $j \neq i$, conclusion (2) holds.

Conclusion 1:

Assume for the sake of contradiction that for both $i \in \{1, 2\}$ there exists some configuration C with $C(X) = X_{hard,\mathcal{N}}$, with $\|C(Y)\|_1 \geq 1$, and with

$$\mathbb{P}[a_i^{t+1} = 1 | N^t = C] < 1 - \frac{8 \cdot \ln 2t_c}{t_s}.$$

From this assumption we can deduce two claims. First, since all y_i are excitatory:

Claim 5.4.21. *For both* $i \in \{1, 2\}$*, there exists some configuration* C *with* $C(X) = X_{hard,\mathcal{N}}$*, with* $\|C(Y)\|_1 = 1$*, and with* $\mathbb{P}[a_i^{t+1} = 1 | N^t = C] < 1 - \frac{8 \cdot \ln 2t_c}{t_s}$*.*

Further, since by Definition 5.4.2, a symmetric SNN has $w(y_i, u) = w(y_j, u)$ for all i, j and $u \in A$, and since each inhibitor can only have incoming connections from the inputs and outputs,

Claim 5.4.22. *For both* $i \in \{1, 2\}$*,* for every C *with* $C(X) = X_{hard,\mathcal{N}}$ *and with* $C(Y)$ *a valid WTA output configuration,*

$$\mathbb{P}[a_i^{t+1} = 1 | N^t = C] < 1 - \frac{8 \cdot \ln 2t_c}{t_s}.$$

Since conditioned on N^t, a_1^{t+1} and a_2^{t+2} are independent, we can compute, for any

C with $C(Y)$ a valid WTA output configuration,

$$\begin{aligned}\mathbb{P}[\|A^{t+1}\|_1 = 1|N^t = C] &= \mathbb{P}[a_1^{t+1} = 1|N^t = C] \cdot \mathbb{P}[a_2^{t+1} = 0|N^t = C] \\ &\quad + \mathbb{P}[a_1^{t+1} = 0|N^t = C] \cdot \mathbb{P}[a_2^{t+1} = 1|N^t = C] \\ &= \mathbb{P}[a_1^{t+1} = 1|N^t = C] \cdot \left(1 - \mathbb{P}[a_2^{t+1} = 1|N^t = C]\right) \\ &\quad + \left(1 - \mathbb{P}[a_1^{t+1} = 1|N^t = C]\right) \cdot \mathbb{P}[a_2^{t+1} = 1|N^t = C]. \quad (5.53)\end{aligned}$$

If $\mathbb{P}[a_1^{t+1} = 1|N^t = C] \geq \frac{1}{2}$, then (5.53) is maximized by setting $\mathbb{P}[a_2^{t+1} = 1|N^t = C] = 0$, giving via Claim 5.4.22:

$$\mathbb{P}[\|A^{t+1}\|_1 = 1|N^t = C] = \mathbb{P}[a_1^{t+1} = 1|N^t = C] < 1 - \frac{8 \ln 2t_c}{t_s}.$$

If $\mathbb{P}[a_1^{t+1} = 1|N^t = C] \leq \frac{1}{2}$, (5.53) is maximized by setting $Pr[a_2^{t+1} = 1|N^t = C] = 1 - \frac{8 \ln 2t_c}{t_s}$, giving:

$$\begin{aligned}\mathbb{P}[\|A^{t+1}\|_1 = 1|N^t = C] &< \mathbb{P}[a_1^{t+1} = 1|N^t = C] \cdot \frac{8 \ln 2t_c}{t_s} \\ &\quad + \left(1 - \mathbb{P}[a_1^{t+1} = 1|N^t = C]\right) \cdot \left(1 - \frac{8 \ln 2t_c}{t_s}\right). \quad (5.54)\end{aligned}$$

Using our requirement that $\frac{8 \ln 2t_c}{t_s} \leq \frac{1}{4}$, (5.54) is maximized by setting $\mathbb{P}[a_1^{t+1} = 1|N^t = C] = 0$, again giving:

$$\mathbb{P}[\|A^{t+1}\|_1 = 1|N^t = C] = \mathbb{P}[a_1^{t+1} = 1|N^t = C] < 1 - \frac{8 \ln 2t_c}{t_s}.$$

In either case, we have a contradiction of Lemma 5.4.18, giving the result.

Conclusion 2:

From conclusion (1) proven above, we can assume without loss of generality that, for any configuration C with $C(X) = X_{hard,\mathcal{N}}$ and with $\|C(Y)\|_1 \geq 1$,

$$\mathbb{P}[a_1^{t+1} = 1|N^t = C] \geq 1 - \frac{8 \cdot \ln 2t_c}{t_s}. \quad (5.55)$$

I.e., we assume that the index i in the lemma statement satisfies $i = 1$. To prove conclusion (2) Assume for the sake of contradiction that there exists some configuration C with $C(X) = X_{hard,\mathcal{N}}$ and with $\|C(Y)\| \leq 1$ such that $\mathbb{P}[a_2^{t+1} = 1|N^t = C] > \frac{12 \cdot \ln 2t_c}{t_s}$. Since all outputs are excitatory, from this assumption we can conclude:

Claim 5.4.23. *There exists some configuration C with $C(X) = X_{hard,\mathcal{N}}$ and with $\|C(Y)\|_1 = 1$ such that $\mathbb{P}[a_2^{t+1} = 1|N^t = C] > \frac{12\cdot\ln 2t_c}{t_s}$.*

Further, since, by Definition 5.4.2, a symmetric SNN has $w(y_i, u) = w(y_j, u)$ for all i, j and $u \in A$, and since each inhibitor can only have incoming connections from the inputs and outputs,

Claim 5.4.24. *For* every C with $C(X) = X_{hard,\mathcal{N}}$ and with $C(Y)$ a valid WTA output configuration,

$$\mathbb{P}[a_2^{t+1} = 1|N^t = C] > \frac{12\cdot\ln 2t_c}{t_s}.$$

Using Claim 5.4.24 we can prove conclusion (2) by considering two cases.

Case 1: $\mathbb{P}[a_2^{t+1} = 1|N^t = C] > 1/2$.

In this case, by (5.53) and (5.55), we have for C with $C(Y)$ a valid WTA output configuration:

$$\begin{aligned}\mathbb{P}[\|A^{t+1}\|_1 = 1|N^t = C] &= \mathbb{P}[a_1^{t+1} = 1|N^t = C]\cdot\left(1 - \mathbb{P}[a_2^{t+1} = 1|N^t = C]\right)\\ &\quad + \left(1 - \mathbb{P}[a_1^{t+1} = 1|N^t = C]\right)\cdot\mathbb{P}[a_2^{t+1} = 1|N^t = C]\\ &< \frac{1}{2} + \frac{8\cdot\ln 2t_c}{t_s}\\ &< 1 - \frac{8\cdot\ln 2t_c}{t_s} \end{aligned}\tag{5.56}$$

where the last inequality follows from our requirement that $\frac{8\cdot\ln 2t_c}{t_s} \leq \frac{1}{4}$. (5.56) contradicts Lemma 5.4.18, giving the result in this case.

Case 2: $\mathbb{P}[a_2^{t+1} = 1|N^t = C] \leq 1/2$.

In this case, by (5.55), Claim 5.4.24, and (5.53) we have for any C with $C(Y)$ a valid WTA output configuration:

$$\begin{aligned}\mathbb{P}[\|A^{t+1}\|_1 = 1|N^t = C] &= \mathbb{P}[a_1^{t+1} = 1|N^t = C]\cdot\left(1 - \mathbb{P}[a_2^{t+1} = 1|N^t = C]\right)\\ &\quad + \left(1 - \mathbb{P}[a_1^{t+1} = 1|N^t = C]\right)\cdot\mathbb{P}[a_2^{t+1} = 1|N^t = C]\\ &< \left(1 - \frac{12\ln 2t_c}{t_s}\right) + \frac{4\cdot\ln 2t_c}{t_s}\\ &< 1 - \frac{8\cdot\ln 2t_c}{t_s}. \end{aligned}\tag{5.57}$$

Again, (5.57) contradicts Lemma 5.4.18, giving the result in this case. □

From Lemma 5.4.20 we can easily show that it is unlikely that a_2 ever fires when a_1 does not.

Corollary 5.4.25 (Neuron a_2 Rarely Fires Alone)**.** *Let $\mathcal{N} = \langle N, w, b, f \rangle$ be any symmetric SNN with two auxiliary inhibitory neurons a_1, a_2 which solves $WTA(n, t_c, t_s, \delta)$ for $n \geq 341$, $\delta \leq 1/2$, $t_c \leq \frac{\ln n}{30 \ln \ln n}$, and $\frac{8 \ln 2t_c}{t_s} \leq \frac{1}{4}$. For any configuration C with $C(X) = X_{hard,\mathcal{N}}$ (Definition 5.4.15):*

$$\mathbb{P}[a_1^{t+1} = 0 \textit{ and } a_2^{t+1} = 1 | N^t = C] \leq \frac{12 \cdot \ln 2t_c}{t_s}.$$

Proof. We prove this result in two cases.

Case 1: $\|C(Y)\|_1 \leq 1$.

In this case, using Lemma 5.4.20 conclusion (2):

$$\mathbb{P}[a_1^{t+1} = 0 \text{ and } a_2^{t+1} = 1 | N^t = C] \leq \mathbb{P}[a_2^{t+1} = 1 | N^t = C] \leq \frac{12 \cdot \ln 2t_c}{t_s}.$$

Case 2: $\|C(Y)\|_1 > 1$.

In this case, applying Lemma 5.4.20 conclusion (1):

$$\begin{aligned}\mathbb{P}[a_1^{t+1} = 0 \text{ and } a_2^{t+1} = 1 | N^t = C] &\leq \mathbb{P}[a_1^{t+1} = 0 | N^t = C] \\ &= 1 - \mathbb{P}[a_1^{t+1} = 0 | N^t = C] \\ &\leq \frac{8 \cdot \ln 2t_c}{t_s}.\end{aligned}$$

□

In combination, Lemma 5.4.16 and Corollary 5.4.25 show that, in order for a valid WTA output state to be maintained with high probability, some inhibitor, (which we assume without loss of generality is a_1) must fire alone. Using this fact, we can characterize the firing behavior of the outputs when a_1 fires alone. Since a_1 maintains stability, if it fires alone at time t, all outputs maintain the same firing state at time $t + 1$ as at time t with high probability.

Lemma 5.4.26 (Neuron a_1 Enforces Stability)**.** *Let $\mathcal{N} = \langle N, w, b, f \rangle$ be any symmetric SNN with two auxiliary inhibitory neurons a_1, a_2 which solves $WTA(n, t_c, t_s, \delta)$ for $n \geq 341$, $\delta \leq 1/2$, $t_c \leq \frac{\ln n}{30 \ln \ln n}$, and $\frac{8 \ln 2t_c}{t_s} \leq \frac{1}{4}$. For any configuration C with $C(X) = X_{hard,\mathcal{N}}$ (Definition 5.4.15) and with $C(a_1) = 1$ and $C(a_2) = 0$:*

1. *For all* $i \in \mathcal{R}$*, if* $C(y_i) = 1$ *then* $\mathbb{P}[y_i^{t+1} = 1 | N^t = C] \geq 1 - \frac{16 \ln 2t_c}{t_s}$*.*

2. *For all* $i \in \mathcal{R}$*, if* $C(y_i) = 0$ *then* $\mathbb{P}[y_i^{t+1} = 0 | N^t = C] \geq \left(1 - \frac{16 \ln 2t_c}{t_s}\right)^{1/(|\mathcal{R}|-1)}$

Proof. In a symmetric SNN (Definition 5.4.2) all outputs have identical incoming connections and biases. Thus, to prove the lemma it suffices to show that *there exists a configuration* C with $C(X) = X_{hard,\mathcal{N}}$ and with $C(a_1) = 1$ and $C(a_2) = 0$, such that:

1. There exists $i \in \mathcal{R}$ with $C(y_i) = 1$ and $\mathbb{P}[y_i^{t+1} = 1 | N^t = C] \geq 1 - \frac{16 \ln 2t_c}{t_s}$.

2. There exists $i \in \mathcal{R}$ with $C(y_i) = 0$ and $\mathbb{P}[y_i^{t+1} = 0 | N^t = C] \geq \left(1 - \frac{16 \ln 2t_c}{t_s}\right)^{1/(|\mathcal{R}|-1)}$.

We will show that these bounds must hold for at least one configuration in order for the network to reach a valid WTA output configuration in t_c steps and remain in this configuration for t_s consecutive steps. Let $\mathcal{E}(t)$ denote the event that $Y^t = Y^{t+1} = Y^{t+2}$. For any C we have:

$$\begin{aligned}
\mathbb{P}[\mathcal{E}(t) | N^t = C] &= \mathbb{P}[\mathcal{E}(t) | N^t = C, a_1^{t+1} = a_2^{t+1}] \cdot \mathbb{P}[a_1^{t+1} = a_2^{t+1} | N^t = C] \\
&\quad + \mathbb{P}[\mathcal{E}(t) | N^t = C, a_1^{t+1} = 0, a_2^{t+1} = 1] \cdot \mathbb{P}[a_1^{t+1} = 0, a_2^{t+1} = 1 | N^t = C] \\
&\quad + \mathbb{P}[\mathcal{E}(t) | N^t = C, a_1^{t+1} = 1, a_2^{t+1} = 0] \cdot \mathbb{P}[a_1^{t+1} = 1, a_2^{t+1} = 0 | N^t = C] \\
&\leq \mathbb{P}[a_1^{t+1} = 0, a_2^{t+1} = 1 | N^t = C] \\
&\quad + \max\left(\mathbb{P}[\mathcal{E}(t) | N^t = C, a_1^{t+1} = a_2^{t+1}], \mathbb{P}[\mathcal{E}(t) | Y^t = C, a_1^{t+1} = 1, a_2^{t+1} = 0]\right). \qquad (5.58)
\end{aligned}$$

We can upper bound the first term of (5.58) by $\frac{12 \ln 2t_c}{t_s}$ using Corollary 5.4.25. Additionally, using Lemma 5.4.16, we can bound $\mathbb{P}[\mathcal{E}(t) | N^t = C, a_1^{t+1} = a_2^{t+1}] \leq \frac{1}{\ln n}$. This gives:

$$\mathbb{P}[\mathcal{E}(t) | N^t = C] \leq \frac{12 \ln 2t_c}{t_s} + \max\left(\frac{1}{\ln n}, \mathbb{P}[\mathcal{E}(t) | N^t = C, a_1^{t+1} = 1, a_2^{t+1} = 0]\right). \qquad (5.59)$$

Applying Lemma 5.4.17 gives that there must exist some C with $C(Y)$ a valid WTA output configuration such that $\mathbb{P}[\mathcal{E}(t) | N^t = C] \geq 1 - \frac{4 \ln 2t_c}{t_s}$. So combining with (5.59):

$$\max\left(\frac{1}{\ln n}, \mathbb{P}[\mathcal{E}(t) | N^t = C, a_1^{t+1} = 1, a_2^{t+1} = 0]\right) \geq 1 - \frac{16 \ln 2t_c}{t_s}$$

By our assumption that $\frac{8\ln 2t_c}{t_s} \leq \frac{1}{4}$ and the fact that $n \geq 341$ we easily have that $\frac{1}{\ln n} \leq \frac{1}{2} \leq 1 - \frac{16\ln 2t_c}{t_s}$, meaning that in fact, there must exist some C with $C(Y)$ a valid WTA output configuration such that:

$$\mathbb{P}[\mathcal{E}(t)|N^t = C, a_1^{t+1} = 1, a_2^{t+1} = 0] \geq 1 - \frac{16\ln 2t_c}{t_s}. \tag{5.60}$$

We can now easily prove the two conclusions of the lemma using similar arguments.

Conclusion 1:

Assume for the sake of contradiction that for every configuration C with $C(X) = X_{hard,\mathcal{N}}$ and with $C(a_1) = 1$ and $C(a_2) = 0$, for all $i \in \mathcal{R}$ with $C(y_i) = 1$,

$$\mathbb{P}[y_i^{t+1} = 1|N^t = C] < 1 - \frac{16\ln 2t_c}{t_s}.$$

Then, for every C with $C(Y)$ a valid WTA output configuration we have:

$$\begin{aligned}\mathbb{P}[\mathcal{E}(t)|N^t = C, a_1^{t+1} = 1, a_2^{t+1} = 0] &\leq \mathbb{P}[Y^{t+2} = Y^{t+1}|N^{t+1} = C, a_1^{t+1} = 1, a_2^{t+1} = 0]\\ &< 1 - \frac{16\ln 2t_c}{t_s}.\end{aligned}$$

contradicting (5.60) and giving the lemma.

Conclusion 2:

Assume for the sake of contradiction that for every configuration C with $C(X) = X_{hard,\mathcal{N}}$ and with $C(a_1) = 1$ and $C(a_2) = 0$, for all $i \in \mathcal{R}$ with $C(y_i) = 0$,

$$\mathbb{P}[y_i^{t+1} = 0|N^t = C] < \left(1 - \frac{16\ln 2t_c}{t_s}\right)^{1/(|\mathcal{R}|-1)}.$$

Then, again for every C with $C(Y)$ a valid WTA output configuration we have:

$$\begin{aligned}\mathbb{P}[\mathcal{E}(t)|N^t = C, a_1^{t+1} = 1, a_2^{t+1} = 0] &\leq \mathbb{P}[Y^{t+2} = Y^{t+1}|N^{t+1} = C, a_1^{t+1} = 1, a_2^{t+1} = 0]\\ &< \left(\left(1 - \frac{16\ln 2t_c}{t_s}\right)^{1/(|\mathcal{R}|-1)}\right)^{|\mathcal{R}|-1}\\ &< 1 - \frac{16\ln 2t_c}{t_s}\end{aligned}$$

contradicting (5.60) and giving the lemma. □

Lemma 5.4.26 implies that when just a_1 fires, since this leads to stability, it is unlikely to lead to a valid WTA output configuration when starting from an invalid output configuration. That is, a_1 does not drive convergence to a valid WTA output configuration.

Corollary 5.4.27 (Neuron a_1 Does Not Drive Convergence)**.** *Let $\mathcal{N} = \langle N, w, b, f\rangle$ be any symmetric SNN with two auxiliary inhibitory neurons a_1, a_2 which solves $WTA(n, t_c, t_s, \delta)$ for $n \geq 341$, $\delta \leq 1/2$, $t_c \leq \frac{\ln n}{30 \ln \ln n}$, and $\frac{8 \ln 2t_c}{t_s} \leq \frac{1}{4}$. For any configuration C with $C(X) = X_{hard,\mathcal{N}}$ (Definition 5.4.15), with $C(a_1) = 1$ and $C(a_2) = 0$, and in which $C(Y)$ is* not a valid WTA output configuration*:*

$$\mathbb{P}[Y^{t+1} \textit{ is a valid WTA output configuration } | N^t = C] \leq \frac{32 \ln 2t_c}{t_s}.$$

Proof. The proof can be split into two cases, depending on if $C(Y)$ is not a valid WTA output configuration because too many valid outputs are firing or because no outputs are firing. Let $\mathcal{R} \subseteq \{1, ...,\}$ be the set of indices with $X_{hard,\mathcal{N}}(x_i) = 1$.

Case 1: $|\{i \in \mathcal{R} | C(y_i) = 1\}| \geq 2$**.**

Let $Y(\mathcal{R})$ denote the set of outputs restricted to the indices in $\mathcal{R}$. In order for Y^{t+1} to be a valid WTA output configuration for $X_{hard,\mathcal{N}}$, we must have $\|Y(\mathcal{R})^t\|_1 = 1$. We can apply Lemma 5.4.26 conclusion (1) to bound:

$$\mathbb{P}[\|Y(\mathcal{R})^{t+1}\|_1 \geq 2 | N^t = C] \geq \left(1 - \frac{16 \ln 2t_c}{t_s}\right)^2 \geq 1 - \frac{32 \ln 2t_c}{t_s}$$

and thus $\mathbb{P}[Y^{t+1}$ is a valid WTA output configuration $|N^t = C] \leq \frac{32 \ln 2t_c}{t_s}$.

Case 2: $|\{i \in \mathcal{R} | C(y_i) = 1\}| = 0$**.**

In this case, we can apply Lemma 5.4.26 conclusion (2) to bound:

$$\mathbb{P}[\|Y(\mathcal{R})^{t+1}\|_1 = 0 | N^t = C] \geq \left(\left(1 - \frac{16 \ln 2t_c}{t_s}\right)^{1/(|\mathcal{R}|-1)}\right)^{|\mathcal{R}|}$$

By our assumption that $n \geq 341$, $\frac{|\mathcal{R}|}{|\mathcal{R}|-1} = \frac{\lfloor \ln^2 n \rfloor}{\lfloor \ln^2 n \rfloor - 1} \leq \frac{34}{33} < 2$. So we have:

$$\mathbb{P}[\|Y(\mathcal{R})^{t+1}\|_1 = 0 | N^t = C] \geq \left(1 - \frac{16 \ln 2t_c}{t_s}\right)^2 \geq 1 - \frac{32 \ln 2t_c}{t_s}.$$

Thus, $\mathbb{P}[Y^{t+1}$ is a valid WTA output configuration $|N^t = C] \leq \frac{32\ln 2t_c}{t_s}$, giving the lemma.

□

We can now prove the main two-inhibitor lower bound Theorem 5.4.14 via a case analysis which combines Lemma 5.4.16, Corollary 5.4.25, and Corollary 5.4.27.

Proof of Theorem 5.4.14.

Assume that $\mathcal{N} = \langle N, w, b, f\rangle$ is a symmetric SNN with two auxiliary inhibitory neurons a_1, a_2 which solves $WTA(n, t_c, t_s, \delta)$ for $n \geq 341$, $\delta \leq 1/2$, $t_c \leq \frac{\ln n}{30 \ln\ln n}$, and $\frac{\ln 2t_c}{t_s} \leq \frac{1}{32\ln n}$. Assume that the network is given the hard input execution $\alpha_{hard,\mathcal{N}}$ (Definition 5.4.15) and starts with initial state N^0 in which Y^0 is not a valid WTA output configuration and $a_1^0 = a_2^0 = 0$.

Let $\mathcal{E}_{fail}(t)$ be the event that for every $t' \leq t$, $Y^{t'}$ is not a valid WTA output configuration. Let $\mathcal{E}_{01}(t)$ be the event that for every $t' \leq t$, we *do not have* $a_1^{t'} = 0$ and $a_2^{t'} = 1$.

We can bound $\mathbb{P}[\mathcal{E}_{fail}(t)] \geq \mathbb{P}[\mathcal{E}_{fail}(t), \mathcal{E}_{01}(t)]$ and write, using that $\mathcal{E}_{fail}(t-1) \subseteq \mathcal{E}_{fail}(t)$ and $\mathcal{E}_{01}(t-1) \subseteq \mathcal{E}_{01}(t)$,

$$\mathbb{P}[\mathcal{E}_{fail}(t), \mathcal{E}_{01}(t)] = \mathbb{P}[\mathcal{E}_{fail}(t), \mathcal{E}_{01}(t)|\mathcal{E}_{fail}(t-1), \mathcal{E}_{01}(t-1)] \cdot \mathbb{P}[\mathcal{E}_{fail}(t-1), \mathcal{E}_{01}(t-1)]. \tag{5.61}$$

Conditioned on $\mathcal{E}_{fail}(t-1)$ and $\mathcal{E}_{01}(t-1)$, Y^{t-1} is not a valid WTA output configuration and $(a_1^{t-1}, a_2^{t-1}) \neq (0,1)$. Let $\mathcal{C}$ be the set of all configurations which are not valid WTA output configurations and which have $(C(a_1), C(a_2)) \neq (0,1)$. We can expand (5.61) as:

$$\begin{aligned}\mathbb{P}[\mathcal{E}_{fail}(t), \mathcal{E}_{01}(t)] &= \mathbb{P}[\mathcal{E}_{fail}(t), \mathcal{E}_{01}(t)|\mathcal{E}_{fail}(t-1), \mathcal{E}_{01}(t-1)] \cdot \mathbb{P}[\mathcal{E}_{fail}(t-1), \mathcal{E}_{01}(t-1)]\\ &= \sum_{C\in\mathcal{C}} \mathbb{P}[\mathcal{E}_{fail}(t), \mathcal{E}_{01}(t)|N^{t-1} = C] \cdot \mathbb{P}[N^{t-1} = C, \mathcal{E}_{fail}(t-1), \mathcal{E}_{01}(t-1)].\end{aligned} \tag{5.62}$$

Note that by the law of total probability since $\mathcal{E}_{fail}(t-1)$ and $\mathcal{E}_{01}(t-1)$ requires that $C \in \mathcal{C}$:

$$\sum_{C\in\mathcal{C}} \mathbb{P}[N^{t-1} = C, \mathcal{E}_{fail}(t-1), \mathcal{E}_{01}(t-1)] = \mathbb{P}[\mathcal{E}_{fail}(t-1), \mathcal{E}_{01}(t-1)]. \tag{5.63}$$

We now bound $\mathbb{P}[\mathcal{E}_{fail}(t), \mathcal{E}_{01}(t)|N^{t-1} = C]$ in (5.62) in two cases:

Case 1: $C(a_1) = C(a_2) = 1$ or $C(a_1) = C(a_2) = 0$.

In this case, for any $C \in \mathcal{C}$ with $C(a_1) = C(a_2)$, by Lemma 5.4.16,

$$\mathbb{P}[\mathcal{E}_{fail}(t)|N^{t-1} = C] \geq 1 - \frac{1}{\ln n}.$$

By Corollary 5.4.25,

$$\mathbb{P}[\mathcal{E}_{01}(t)|N^{t-1} = C] \geq 1 - \frac{12 \cdot \ln 2t_c}{t_s} \geq 1 - \frac{1}{\ln n},$$

where the second inequality follows from our requirement that $\frac{\ln 2t_c}{t_s} \leq \frac{1}{32 \ln n}$. Additionally, conditioned on N^{t-1}, $\mathcal{E}_{fail}(t)$ and $\mathcal{E}_{01}(t)$ are independent since they involve disjoint sets of neurons. Thus we can bound:

$$\mathbb{P}[\mathcal{E}_{fail}(t), \mathcal{E}_{01}(t)|N^{t-1} = C] \geq \left(1 - \frac{1}{\ln n}\right)^2. \tag{5.64}$$

Case 2: $C(a_1) = 1$ and $C(a_2) = 0$.

In this case, for any $C \in \mathcal{C}$, since $C(Y)$ is not a valid WTA output configuration, by Corollary 5.4.27,

$$\mathbb{P}[\mathcal{E}_{fail}(t)|N^{t-1} = C] \geq 1 - \frac{32 \cdot \ln 2t_c}{t_s} \geq 1 - \frac{1}{\ln n}$$

where the second inequality follows from our requirement that $\frac{\ln 2t_c}{t_s} \leq \frac{1}{32 \ln n}$. As above we also have $\mathbb{P}[\mathcal{E}_{01}(t)|N^{t-1} = C] \geq 1 - \frac{1}{\ln n}$ and since $\mathcal{E}_{fail}(t)$ and $\mathcal{E}_{01}(t)$ are independent conditioned on N^{t-1}:

$$\mathbb{P}[\mathcal{E}_{fail}(t), \mathcal{E}_{01}(t)|N^{t-1} = C] \geq \left(1 - \frac{1}{\ln n}\right)^2. \tag{5.65}$$

Together (5.64) and (5.65) and the definition of $\mathcal{C}$ give us:

Claim 5.4.28. *For any C which is not a valid WTA configuration and with $(C(a_1), C(a_2)) \neq (0, 1)$,*

$$\mathbb{P}[\mathcal{E}_{fail}(t), \mathcal{E}_{01}(t)|N^{t-1} = C] \geq \left(1 - \frac{1}{\ln n}\right)^2.$$

Completing the Theorem.

We conclude by using Claim 5.4.28 to lower bound the probability of converging to a valid WTA configuration within t_c steps. Substituting the bound of Claim 5.4.28 into (5.62) and (5.63) we have:

$$\begin{aligned}\mathbb{P}[\mathcal{E}_{fail}(t), \mathcal{E}_{01}(t)] &\geq \sum_{C \in \mathcal{C}} \left(1 - \frac{1}{\ln n}\right)^2 \cdot \mathbb{P}[N^{t-1} = C, \mathcal{E}_{fail}(t-1), \mathcal{E}_{01}(t-1)] \\ &\geq \left(1 - \frac{1}{\ln n}\right)^2 \cdot \mathbb{P}[\mathcal{E}_{fail}(t-1), \mathcal{E}_{01}(t-1)]\end{aligned}$$

Since we choose N^0 with Y^0 not a valid WTA output configuration and $a_1^0 = a_2^0 = 0$, $\mathbb{P}[\mathcal{E}_{fail}(0), \mathcal{E}_{01}(0)] = 1$. Thus, via induction, we have for any $t \geq 0$,

$$\mathbb{P}[\mathcal{E}_{fail}(t), \mathcal{E}_{01}(t)] \geq \left(1 - \frac{1}{\ln n}\right)^{2t}.$$

Plugging in $t_c \leq \frac{\ln n}{30 \ln \ln n}$ we have:

$$\begin{aligned}\mathbb{P}[\mathcal{E}_{fail}(t_c)] \geq \mathbb{P}[\mathcal{E}_{fail}(t_c), \mathcal{E}_{01}(t_c)] &\geq \left(1 - \frac{1}{\ln n}\right)^{\frac{\ln n}{15 \ln \ln n}} \\ &\geq \frac{1}{e^{1/15}} > 1/2.\end{aligned}$$

This contradicts the fact that $\mathcal{N}$ solves $WTA(n, t_c, t_s, \delta)$ for $\delta \geq 1/2$, giving the lower bound. □

5.5 Faster Convergence With More Inhibitors

In this section we show how to speed up the two-inhibitor construction of Section 5.3 by using $\alpha > 2$ inhibitors. We give a formal convergence proof for a construction which uses $\lceil \log_2 n \rceil + 1$ inhibitors and converges with constant probability in $O(1)$ time (and with probability $\geq 1 - \delta$ in $O(\log 1/\delta)$ time). We then describe the high level idea behind two constructions that give a tradeoff between the number of inhibitors used and the convergence time.

5.5.1 Use of History Period

Our $\lceil \log_2 n \rceil + 1$-inhibitor construction (as well as the sketched constructions which give an inhibitor-convergence time tradeoff) requires using a history period of $h = 2$, as suggested in Section 5.2.5. At a high level, to achieve fast convergence, our networks use the larger number of inhibitors available to create higher levels of inhibition at time t corresponding to higher number of firing outputs at time $t-1$. This strategy however, leads to 'race conditions'. If many outputs fire at time t and exactly one output fires at time $t+1$, there will still be a high level of inhibition at time $t+1$. Thus, at time $t+2$ it is likely that the single firing output at time $t+1$ will stop firing and so the network will not stabilize to a valid WTA output state.

To avoid this race condition, we use history so that an output's self-loop excites the output for two time steps. The stability inhibitor a_s will also be excited for two time steps by the outputs. In this way, if a single output y_i fires at time t, while at time $t+1$ no outputs may fire due to high levels of inhibition, at time $t+2$ the history will cause both a_s and y_i to fire, and the network to stabilize to a valid WTA state.

Generalized Model with History Period

Following the basic SNN model of Section 5.2 we define an SNN model with history period h for any $h \geq 1$. The changes to the definition are highlighted in gray.

An SNN $\mathcal{N} = \langle N, w, b, f, h \rangle$ with history period h consists of:

- N, a set of neurons, partitioned into a set of input neurons X, a set of output neurons Y, and a set of auxiliary neurons A. N is also partitioned into a set of excitatory and inhibitory neurons E and I. All input and output neurons are excitatory.
- $h \in \mathbb{Z}^{\geq 1}$, a positive integer indicating the neural response history period.
- $w : N \times N \times \{1, ..., h\} \rightarrow \mathbb{R}$, a weight function describing the weighted synaptic connections between the neurons in the network. w is restricted in a few notable ways:
  - $w(u, x, l) = 0$ for all $u \in N$, $x \in X$, and $l \in \{1, ..., h\}$.
  - Each excitatory neuron $v \in E$ has $w(v, u, l) \geq 0$ for every u and $l \in \{1, ..., h\}$. Each inhibitory neuron $v \in I$ has $w(v, u, l) \leq 0$ for every u and $l \in \{1, ..., h\}$.
- $b : N \rightarrow \mathbb{R}$, a bias function, assigning an activation bias to each neuron.
- $f : \mathbb{R} \rightarrow [0, 1]$, a spike probability function, satisfying a few restrictions:
  - f is continuous and monotonically increasing.
  - $\lim_{x\rightarrow\infty} f(x) = 1$ and $\lim_{x\rightarrow-\infty} f(x) = 0$.

Remark on the Time Dependent Weight Function: The only difference between the above model and our basic SNN model of Section 5.2.1 is in the specification of the weight function w. In the model with history period h, w describes the strength of the synaptic connections between neurons in N, as a function of the time difference between a spike and the current time (from 1 up to h). $w(u, v, 1)$ is the weight corresponding to the most recent time and $w(u, v, h)$ corresponds to the most distant time within the history period. The weight function, for example, can be used to model the decaying effect of a spike over time, if we set $|w(u, v, 1)| \geq |w(u, v, 2)| \geq ... \geq |w(u, v, h)|$.

Network Dynamics With History Period

In our SNN with history period model, configurations and executions are defined as in our basic model (see Section 5.2.2). The behavior of the SNN is determined as follows:

- **Input Neurons**: As in our basic model, we specify how the infinite input execution $X^0X^1....$ is determined. In this chapter, we will fix the input so that for each $u \in X$, u^t is constant for all $t \geq 0$.

- **Initial Firing States**: For each non-input $u \in N \setminus X$, the firing states in the first h time slots $u^0, u^1, ..., u^{h-1}$ are arbitrary, where h is this history length.

- **Firing Dynamics**: For each non-input neuron $u \in N \setminus X$ and every time $t \geq h$, let $\text{pot}(u, t)$ denote the membrane potential at time t and $p(u, t)$ denote the corresponding firing probability. These values are calculated as:

$$\text{pot}(u,t) = \left(\sum_{i=1}^{h} \sum_{v \in N} w(v, u, i) \cdot v^{t-i} \right) - b(u) \text{ and } p(u, t) = f(\text{pot}(u, t)) \quad (5.66)$$

  where f is the spike probability function. At time t, each non-input neuron u fires independently with probability $p(u, t)$. Note that equation (5.66) is defined only for $t \geq h$, in which case $t - i \geq 0$ for all $i \in \{1, ..., h\}$. It is analogous to the potential calculation (5.1), used for our basic model, except that the summation of spikes is over h time steps.

An SSN $\mathcal{N} = \langle N, w, b, f, h \rangle$ with history period h, length h execution $\alpha^h = N^0N^1...N^{h-1}$, and infinite input execution α_X define a probability distribution over infinite executions, $\mathcal{D}(\mathcal{N}, \alpha^h, \alpha_X)$. This distribution is the natural distribution that follows from applying the stochastic firing dynamics of (5.66). As with our basic model (see Section 5.2.2), we can also define a corresponding distribution $\mathcal{D}_Y(\mathcal{N}, \alpha^h, \alpha_X)$ on infinite output executions.

Solving Problems in Networks with History

As in our basic model, a problem P is a mapping from an infinite input execution α_X to a set of output distributions. A network $\mathcal{N}$ is said to *solve problem P on input* α_X if, for any length h initial execution α^h, the output distribution $\mathcal{D}_Y(\mathcal{N}, \alpha^h, \alpha_X)$ is an element of $P(\alpha_X)$. A network $\mathcal{N}$ is said to *solve problem P* if it solves P on every infinite input execution α_X.

5.5.2 $O(1)$ Convergence Time with $O(\log n)$ Inhibitors

In this section we describe and analyze a family of networks that converge to a valid WTA state with constant probability in constant time and uses $O(\log n)$ inhibitors.

We have one stability inhibitor a_s that functions similarly to the stability inhibitor in our two-inhibitor construction (Definition 5.3.1), ensuring that, once the network reaches a valid WTA output configuration, it remains in this configuration for t_s consecutive time steps with high probability (see Corollary 5.5.19). This stability inhibitor employs a history period of length 2. We prove in Lemma 5.5.7 that it fires with high probability at time $t+1$ whenever at least one output fires at time t or $t-1$.

We additionally have $\lceil \log_2 n \rceil$ convergence inhibitors, labeled $a_1, ..., a_{\lceil \log_2 n \rceil}$. For each i, a_i fires with high probability at time $t+1$ whenever $k \geq 2^i$ outputs fire at time t. In this way, with high probability, $a_1, ..., a_i$ fire and all other inhibitors do not fires at time $t+1$ whenever the number of firing outputs k is in the range $[2^i, 2^{i+1})$ (see Lemma 5.5.8). Note that we define our network so $a_1, ..., a_{\lceil \log_2 n \rceil}$ have no incoming connections that use the length 2 history period (i.e., $w(u, a_i, 2) = 0$ for all u and all $i \in \{1, ..., \lceil \log_2 n \rceil\}$). Thus, the firing probabilities of these inhibitors at time $t+1$ depend only on the firing pattern at time t.

We set the inhibitory weights such that when a_i fires at time t (along with a_j for all $j \leq i$), each firing output fires at time $t+1$ with probability $p_i \approx \frac{1}{2^i}$ (see Lemma 5.5.12). For $k \in [2^i, 2^{i+1})$ we thus have $p_i \cdot k \in [1, 2)$. We will show that this ensures that, with constant probability, exactly one output fires at time $t+1$ (see Corollary 5.5.13). Once a single output fires, using the length-two history mechanism described in Section 5.5.1, the network stabilizes to a valid WTA state with high probability.

We begin with a formal definition of the network below:

Definition 5.5.1 (Constant Time WTA Network). *For any $n \in \mathbb{Z}^{\geq 2}$ and $\gamma \in \mathbb{R}^+$, let $\mathcal{L}_{n,\gamma} = \langle N, w, b, f, 2\rangle$ where the spike probability, weight, and bias functions are defined as follows:*

- *The spike probability function f is defined to be the basic sigmoid function:*

$$f(x) \overset{\text{def}}{=} \frac{1}{1+e^{-x}}.$$

- *The set of neurons N consists of a set of n input neurons X, labeled $x_1, ..., x_n$, a set of n corresponding outputs Y, labeled $y_1, ..., y_n$, and $\lceil \log_2 n \rceil + 1$ auxiliary inhibitory neurons labeled a_s and $a_1, ..., a_{\lceil \log_2 n \rceil}$.*

- *The weight function w is given by:*

  - *$w(x_i, y_i, 1) = 6\gamma$, for all i.*
  - *$w(y_i, y_i, 1) = w(y_i, y_i, 2) = 2\gamma$, for all i.*
  - *$w(a_s, y_i, 1) = -\gamma$, for all i*
  - *$w(a_1, y_i, 1) = -7\gamma/2 - \ln(2)$, for all i*
  - *$w(a_j, y_i, 1) = -\ln(2)$, for all i and $j \in \{2, ..., \lceil \log_2 n \rceil\}$.*
  - *$w(y_i, a_s, 1) = w(y_i, a_s, 2) = \gamma$, for all i.*
  - *$w(y_i, a_j, 1) = \gamma$, for all i and $j \in \{1, ..., \lceil \log_2 n \rceil\}$.*
  - *$w(u, v, i) = 0$ for any u, v and $i \in \{1, 2\}$ whose connection is not specified above.*

- *The bias function b is given by:*

  - *$b(y_i) = 11\gamma/2$ for all i.*
  - *$b(a_s) = \gamma/2$.*
  - *$b(a_i) = 2^i \cdot \gamma - \gamma/2$ for all $i \in \{1, ..., \lceil \log_2 n \rceil\}$.*

Use of History. Note that in our network definition above, we use the history mechanism only in two places. We set $w(y_i, y_i, 1) = w(y_i, y_i, 2) = 2\gamma$, for all i, meaning that each output's self-loop affects its potential for two time steps. We also set $w(y_i, a_s, 1) = w(y_i, a_s, 2) = \gamma$ for all i, meaning that the stability inhibitor is affected by the outputs for two steps.

Due to the high level of inhibition inducted by the convergence inhibitors $a_1, ..., a_{\lceil \log_2 n \rceil}$, after the network reaches a configuration with just a single firing output, it will likely transition to a state with no firing outputs, since the number of firing inhibitors will still reflect the number of firing outputs in the previous time step. The length-two output self-loop and output to inhibitor connections allow the network to recover from this state. Specifically, in Lemma 5.5.18 and Corollary 5.5.19, we show that if the network reaches a valid WTA output state at time t, with good probability it will return to this state at time $t + 2$ and remain in this state for t_s consecutive steps.

In our two-inhibitor construction, no history was necessary. The inhibition in the network was always low enough such that, after reaching a near-valid WTA configuration with a single firing output, with constant probability the network would transition to a valid-WTA configuration and stabilize in this configuration (see Lemma 5.3.21).

We prove the following theorems on the performance of $\mathcal{L}_{n,\gamma}$:

Theorem 5.5.2 ($O(\log n)$-Inhibitor WTA)**.** *For* $\gamma \geq \ln((n+2)t_s/\delta)$*,* $\mathcal{L}_{n,\gamma}$ *solves* $\text{WTA}(n, t_c, t_s, \delta)$ *for any* $t_c \geq \log_2(1/\delta) + 1$*.* $\mathcal{L}_{n,\gamma}$ *contains* $\lceil \log_2 n \rceil + 1$ *auxiliary inhibitors.*

Theorem 5.5.3 ($O(\log n)$-Inhibitor Expected-Time WTA)**.** *For* $\gamma \geq \ln((n+2)t_s/\delta)$*,* $\mathcal{L}_{n,\gamma}$ *solves* $\text{WTA-EXP}(n, t_c, t_s)$ *for any* $t_c \geq 4$*.* $\mathcal{L}_{n,\gamma}$ *contains* $\lceil \log_2 n \rceil + 1$ *auxiliary inhibitors.*

Proof Roadmap. We prove Theorems 5.5.2 and 5.5.3 in Section 5.5.5. The analysis is broken down as follows:

> Section 5.5.3: Prove basic *two-step lemmas* which characterize single time step transitions of $\mathcal{L}_{n,\gamma}$, showing that the neurons behave as described in the above high-level description.
>
> Section 5.5.4: Prove that, once in a valid WTA output configuration, as long as certain other stability conditions are satisfied, $\mathcal{L}_{n,\gamma}$ stays in this configuration with high probability.
>
> Section 5.5.5: Show that all configurations of $\mathcal{L}_{n,\gamma}$ transition with constant probability to a stable and valid WTA configuration within $O(1)$ time steps.
>
> Section 5.5.6: Complete the analysis, demonstrating with what parameter values $\mathcal{L}_{n,\gamma}$ solves the winner-take-all problem (Definitions 5.2.7 and 5.2.9).

5.5.3 Two-Step Lemmas

As in our analysis of our two-inhibitor network family in Section 5.3.2, we begin with a series of lemmas which characterize the basic transition probabilities of $\mathcal{L}_{n,\gamma}$. Since we employ history period $h = 2$, these lemmas consider the behavior of the network at time $t + 1$ conditioned on its configuration at times t and $t - 1$.

We first note that an analog to Lemma 5.3.4 still holds for all inhibitory neurons.

Lemma 5.5.4 (Characterization of Firing Probabilities)**.** *For any time $t \geq 1$ and any $a \in A$:*

$$\begin{aligned}
&\textit{If } \operatorname{pot}(u,t) = 0, \textit{ then } p(u,t) = 1/2.\\
&\textit{If } \operatorname{pot}(u,t) < 0, \textit{ then } p(u,t) \leq e^{-\gamma/2}.\\
&\textit{If } \operatorname{pot}(u,t) > 0, \textit{ then } p(u,t) \geq 1 - e^{-\gamma/2}.
\end{aligned}$$

Proof. The proof is essentially identical to that of Lemma 5.3.4. We just use that for each inhibitor $a \in A$ and each i and $h \in \{1, 2\}$, $w(y_i, a, h)$ and $b(a)$ are all multiples of $\gamma/2$. □

Analogs to Lemma 5.3.5 and Corollary 5.3.6 also still hold, ensuring that, with high probability, outputs that do not correspond to firing inputs do not fire.

Lemma 5.5.5 (Correct Output Behavior)**.** *For any time $t \geq 1$, any configurations C, C' of $\mathcal{L}_{n,\gamma}$, and any i with $C(x_i) = 0$,*

$$\mathbb{P}[y_i^{t+1} = 1 | N^t = C, N^{t-1} = C'] \leq e^{-3\gamma/2}.$$

Proof. If $N^t = C$ then $x_i^t = C(x_i)$. We can compute y_i's potential at time $t + 1$, assuming $x_i^t = 0$:

$$\begin{aligned}
\operatorname{pot}(y_i, t+1) = w(x_i, y_i, 1)x_i^t + w(y_i, y_i, 1)y_i^t + w(y_i, y_i, 2)y_i^{t-1} + w(a_s, y_i, 1)a_s^t\\
+ \sum_{j=1}^{\lceil \log_2 n \rceil} w(a_c, y_i, 1)a_c^t - b(y_i)\\
\leq 0 + 2\gamma + 2\gamma + 0 + 0 - 11\gamma/2 = -3\gamma/2.
\end{aligned}$$

We thus have $p(y_i, t+1) \leq \frac{1}{1+e^{3\gamma/2}} \leq e^{-3\gamma/2}$. □

Corollary 5.5.6 (Correct Output Behavior, All Neurons). *For any time $t \geq 1$ and configurations C, C' of $\mathcal{L}_{n,\gamma}$,*

$$\mathbb{P}[y_i^{t+1} \leq x_i^t \text{ for all } i | N^t = C, N^{t-1} = C'] \geq 1 - ne^{-3\gamma/2}.$$

Proof. The proof is essentially identical to that of Corollary 5.3.6. If $C(x_i) = 1$ then conditioned on $N^t = C$, $x_i^t = 1$ and so $y_i^{t+1} \leq x_i^t$ always. Otherwise, by Lemma 5.5.5, if $C(x_i) = 0$, then $\mathbb{P}[y_i^{t+1} = 0 | N^t = C, N^{t-1} = C'] \geq 1 - e^{-3\gamma/2}$. Union bounding over all such inputs (of which there are at most n) gives the corollary. □

We next show that the inhibitors a_s, $a_1, ..., a_{\lceil \log_2 n \rceil}$ behave as expected. The following lemmas can be viewed as a generalization of Lemma 5.3.7. We first show that, due to our use of history, a_s fires with high probability at time $t+1$ whenever *at least one output fires at time t or $t-1$.*

Lemma 5.5.7 (Correct Stability Inhibitor Behavior). *For any time $t \geq 1$ and configurations C, C' of $\mathcal{L}_{n,\gamma}$,*

1. *If $\|C(Y)\|_1 = \|C'(Y)\|_1 = 0$, then $\mathbb{P}[a_s^{t+1} = 0 | N^t = C, N^{t-1} = C'] \geq 1 - e^{-\gamma/2}$.*
2. *If $\|C(Y)\|_1 \geq 1$ or $\|C'(Y)\|_1 \geq 1$, then $\mathbb{P}[a_s^{t+1} = 1 | N^t = C, N^{t=1} = C'] \geq 1 - e^{-\gamma/2}$.*

Proof. We prove the two conclusions separately.

Conclusion 1: $\|C(Y)\|_1 = \|C'(Y)\|_1 = 0$.

In this case, a_s receives no excitatory signal from the outputs so,

$$\text{pot}(a_s, t+1) = -b(a_s) < 0.$$

Thus by Lemma 5.5.4,

$$\mathbb{P}[a_s^{t+1} = 0 | N^t = C] \geq 1 - e^{-\gamma/2}.$$

Conclusion 2: $\|C(Y)\|_1 \geq 1$ or $\|C'(Y)\|_1 \geq 1$.

In this case we have:

$$\begin{aligned}\text{pot}(a_s, t+1) &= \sum_{j=1}^{n} [w(y_j, a_s, 1) y_j^t + w(y_j, a_s, 2) y_j^{t-1}] - b(a_s) \\ &\geq \gamma - \gamma/2 = \gamma/2.\end{aligned}$$

We thus have by Lemma 5.5.4:

$$\mathbb{P}[a_s^{t+1} = 1 | N^t = C] \geq 1 - e^{-\gamma/2},$$

which gives the lemma. □

As described in Section 5.5.2, the convergence inhibitors $a_1, ..., a_{\lceil \log_2 n \rceil}$ fire at time $t+1$ depending on the number of firing outputs at time t. They have no incoming connections which affect them for two rounds, and thus their firing probabilities at time $t+1$ do not depend on the firing pattern at time $t-1$. We prove that a_j for all $j \leq i$ fire with high probability at time $t+1$ whenever the number of firing outputs at time t falls in the range $[2^i, 2^{i+1})$. Further, all a_j for $j > i$ do not fire with high probability.

Lemma 5.5.8 (Correct Convergence Inhibitor Behaviors). *For any time $t \geq 1$ and configurations C, C' of $\mathcal{L}_{n,\gamma}$,*

1. *If $\|C(Y)\|_1 \leq 1$, then*

$$\mathbb{P}\left[a_i^{t+1} = 0 \textit{ for all } i \in \{1, ..., \lceil \log_2 n \rceil\} \middle| N^t = C, N^{t-1} = C'\right] \geq 1 - \lceil \log_2 n \rceil \cdot e^{-\gamma/2}.$$

2. *For any $i \in \{1, ..., \lceil \log_2 n \rceil\}$, if $\|C(Y)\|_1 = k$ for $k \in [2^i, 2^{i+1})$, then:*

$$\begin{aligned}\mathbb{P}[a_1^{t+1} = ... = a_i^{t+1} = 1 \textit{ and } a_{i+1}^{t+1} = ... = a_{\lceil \log_2 n \rceil}^{t+1} = 0 | N^t = C, N^{t-1} = C'] \\ \geq 1 - \lceil \log_2 n \rceil \cdot e^{-\gamma/2}.\end{aligned}$$

Proof. We prove the two conclusions of the lemma separately.

Conclusion 1: $\|C(Y)\|_1 \leq 1$.

In this case we have for all i:

$$\begin{aligned}\text{pot}(a_i, t+1) &= \sum_{j=1}^{n} w(y_j, a_i, 1) y_j^t - b(a_i) \\ &\leq \gamma - 3\gamma/2 = -\gamma/2.\end{aligned}$$

Again by Lemma 5.5.4 and a union bound,

$$\mathbb{P}\left[\sum_{i=1}^{\lceil \log_2 n \rceil} a_i^{t+1} = 0 \middle| N^t = C, N^{t-1} = C'\right] \geq 1 - \lceil \log_2 n \rceil \cdot e^{-\gamma/2}.$$

Conclusion 2: $\|C(Y)\|_1 = k$ **for** $k \in [2^i, 2^{i+1})$**.**

In this case, for any $j \leq i$ we have:

$$\begin{aligned}\text{pot}(a_j, t+1) &= \sum_{l=1}^{n} w(y_l, a_s, 1) y_l^t - b(a_j) \\ &= k \cdot \gamma - 2^j \gamma + \gamma/2 \geq \gamma/2\end{aligned}$$

where the last inequality follows since $k \geq 2^j \geq 2^j$. In contrast for $j > i$:

$$\begin{aligned}\text{pot}(a_j, t+1) &= \sum_{l=1}^{n} w(y_l, a_s, 1) y_l^t - b(a_j) \\ &= k \cdot \gamma - 2^j \gamma + \gamma/2 \leq -\gamma/2\end{aligned}$$

where the last inequality follows from the fact that $k < 2^{i+1} \leq 2^j$. Overall, by Lemma 5.5.4 and a union bound,

$$\begin{aligned}\mathbb{P}[a_1^{t+1} = ... = a_i^{t+1} = 1 \text{ and } a_{i+1}^{t+1} = ... = a_{\lceil \log_2 n \rceil}^{t+1} = 0 | N^t = C, N^{t-1} = C'] \\ \geq 1 - \lceil \log_2 n \rceil \cdot e^{-\gamma/2},\end{aligned}$$

which gives the lemma □

It will be useful in our future bounds to consider the class of configurations in which all outputs and inhibitors behave as expected. Such a configuration is analogous to the good configurations of our two-inhibitor networks (Definition 5.3.17), from which we were able to show convergence.

Definition 5.5.9 (Typical Configuration)**.** *A* typical configuration *is any configuration C with $C(y_i) \leq C(x_i)$ for all i and $C(a_s) \geq C(a_1) \geq ... \geq C(a_{\lceil \log_2 n \rceil})$.*

In combination, Corollary 5.5.6, Lemma 5.5.7 and Lemma 5.5.8 give:

Corollary 5.5.10 (Correct Behavior, All Neurons)**.** *Assume the input execution α_X of $\mathcal{L}_{n,\gamma}$ has X^t fixed for all t and consider configurations C, C' with $C(X) = C'(X) = X^t$. For any time $t \geq 1$:*

$$\mathbb{P}[N^{t+1} \textit{ is a typical configuration } | N^t = C, N^{t-1} = C'] \geq 1 - (n + \lceil \log_2 n \rceil + 1) \cdot e^{-\gamma/2}.$$

Proof. Since the input is fixed, by Corollary 5.5.6,

$$\mathbb{P}[y_i^{t+1} \leq x_i^{t+1} | N^t = C, N^{t-1} = C'] \geq 1 - n \cdot e^{-3\gamma/2}. \tag{5.67}$$

If $\|C(Y)\|_1 \leq 1$ then by Lemma 5.5.8 conclusion (2),

$$\mathbb{P}[a_1^{t+1} = ... = a_{\lceil \log_2 n \rceil}^{t+1} = 0 | N^t = C, N^{t-1} = C'] \geq 1 - \lceil \log_2 n \rceil \cdot e^{-\gamma/2}.$$

If $\|C(Y)\|_1 > 1$ then by Lemma 5.5.7 conclusion (2), Lemma 5.5.8 conclusion (2), and a union bound,

$$\begin{aligned}\mathbb{P}[a_s^{t+1} = a_1^{t+1} = ... = a_i^{t+1} = 1 \text{ and } a_{i+1}^{t+1} = ... = a_{\lceil \log_2 n \rceil}^{t+1} = 0 | N^t = C, N^{t-1} = C']\\ \geq 1 - (\lceil \log_2 n \rceil + 1) \cdot e^{-\gamma/2}.\end{aligned}$$

Combining these two cases, we have

$$\mathbb{P}[a_s^{t+1} \geq a_1^{t+1} \geq ... \geq a_{\lceil \log_2 n \rceil}^{t+1} | N^t = C, N^{t-1} = C'] \geq 1 - (\lceil \log_2 n \rceil + 1) \cdot e^{-\gamma/2},$$

which gives the lemma after a union bound with (5.67) □

We next show that the stability inhibitor firing alone, with high probability, induces exactly the outputs that fired at one of the previous two time steps to fire in the next step. This Lemma is analogous to Lemma 5.3.9 for our two-inhibitor networks.

Lemma 5.5.11 (Stability Inhibitor Effect)**.** *Assume the input execution α_X of $\mathcal{L}_{n,\gamma}$ has X^t fixed for all t and consider configurations C, C' with $C(X) = C'(X) = X^t$, $C(a_s) = 1$, $C(a_i) = 0$ for all $i \in \{1, ..., \lceil \log_2 n \rceil\}$, and $C(y_i) \leq C(x_i)$, $C'(y_i) \leq C'(x_i)$ for all i. For any time $t \geq 1$,*

$$\mathbb{P}[y_i^{t+1} = \max(y_i^t, y_i^{t-1}) \textit{ for all } i | N^t = C, N^{t-1} = C'] \geq 1 - ne^{-\gamma/2}$$

Proof. Conditioned on $N^t = C, N^{t-1} = C'$, $y_i^t \leq x_i^t$ and $y_i^{t-1} \leq x_i^{t-1}$ by assumption. So for any output with $\max(y_i^t, y_i^{t-1}) = 1$ we must have $x_i^t = 1$. This gives:

$$\begin{aligned}\text{pot}(y_i, t+1) &= w(x_i, y_i, 1)x_i^t + w(y_i, y_i, 1)y_i^t + w(y_i, y_i, 2)y_i^{t-1} + w(a_s, y_i, 1)a_s^t \\ &\qquad + \sum_{j=1}^{\lceil \log_2 n \rceil} w(a_j, y_i, 1)a_j^t - b(y_i) \\ &\geq 6\gamma + 2\gamma \cdot \max(y_i^t, y_i^{t-1}) - \gamma + 0 - 11\gamma/2s \\ &\geq 3\gamma/2.\end{aligned}$$

In contrast, for any output with $\max(y_i^t, y_i^{t-1}) = 0$:

$$\begin{aligned}\text{pot}(y_i, t+1) &= w(x_i, y_i, 1)x_i^t + w(y_i, y_i, 1)y_i^t + w(y_i, y_i, 2)y_i^{t-1} + w(a_s, y_i, 1)a_s^t \\ &\qquad + \sum_{j=1}^{\lceil \log_2 n \rceil} w(a_j, y_i, 1)a_j^t - b(y_i) \\ &\leq 6\gamma + 0 + 0 - \gamma + 0 - 11\gamma/2 \\ &= -\gamma/2.\end{aligned}$$

So, if $\max(y_i^t, y_i^{t-1}) = 1$, then $y_i^{t+1} = 1$ with probability $\geq 1 - e^{-3\gamma/2}$. If $\max(y_i^t, y_i^{t-1}) = 0$, then $y_i^{t+1} = 0$ with probability $\geq 1 - e^{-\gamma/2}$. The lemma follows after union bounding over all n outputs. □

We next characterize the effect of the convergence inhibitors. We show that when l inhibitors fire, any firing output that also fired in the previous time step, fires with probability $\Theta(1/l)$. We will show in Corollary 5.5.13 that this implies that in the next step, with constant probability, exactly one output fires, and in fact the configuration is a valid WTA output configuration.

Lemma 5.5.12 below is analogous to Lemma 5.3.10 for our two-inhibitor networks, except that the firing probability is $\Theta(1/l)$ rather than $1/2$. Since this firing probability is smaller when a larger number of outputs fire at time t, convergence to a single firing output with constant probability occurs in 1 step, rather than $O(\log n)$ steps.

Lemma 5.5.12 (Convergence Inhibitor Effect)**.** *Assume the input execution α_X of $\mathcal{L}_{n,\gamma}$ has X^t fixed for all t and consider configurations C, C' with $C(X) = C'(X) = X^t$, $C(a_s) = C(a_1) = ...C(a_l) = 1$, $C(a_{l+1}) = ... = C(a_{\lceil \log_2 n \rceil}) = 0$ for some $l \geq 1$, and $C(y_i) \leq C(x_i)$, $C'(y_i) \leq C'(x_i)$ for all i. For any time $t \geq 1$,*

1. $\mathbb{P}[y_i^{t+1} \leq \min(y_i^t, y_i^{t-1})$ *for all* $i | N^t = C, N^{t-1} = C'] \geq 1 - ne^{-2\gamma}$.
2. *If* $\min(y_i^t, y_i^{t-1}) = 1$, $\mathbb{P}[y_i^{t+1} = 1 | N^t = C, N^{t-1} = C'] = \frac{1}{1+2^\ell}$.
3. *For* $i \neq j$, y_i^{t+1} *and* y_j^{t+1} *are independent conditioned on* $N^t = C, N^{t-1} = C'$.

Proof. Conditioned on $N^t = C, N^{t-1} = C'$, if $\min(y_i^t, y_i^{t-1}) = 1$, then $y_i^t = y_i^{t-1} = 1$ and by assumption $x_i^t = 1$. We can thus compute:

$$\begin{aligned}\text{pot}(y_i, t+1) &= w(x_i, y_i, 1)x_i^t + w(y_i, y_i, 1)y_i^t + w(y_i, y_i, 2)y_i^{t-1} + w(a_s, y_i, 1)a_s^t + w(a_1, y_i, 1)a_1^t \\ &\quad + \sum_{j=2}^{\lceil \log_2 n \rceil} w(a_j, y_i, 1)a_j^t - b(y_i) \\ &= 6\gamma + 4\gamma - \gamma - 7\gamma/2 - \ln 2 - (l-1) \cdot \ln 2 - 11\gamma/2 \\ &= -l \cdot \ln 2.\end{aligned}$$

We thus have
$$\mathbb{P}[y_i^{t+1} = 1 | N^t = C] = f(-l \cdot \ln 2) = \frac{1}{1+2^l}.$$

This gives conclusion (2). Conclusion (3) holds since, with N^t and N^{t-1} fixed with history length 2, u^{t+1} is independent of v^{t+1} for all $u \neq v$. We can also bound if $\min(y_i^t, y_i^{t-1}) = 0$:

$$\begin{aligned}\text{pot}(y_i, t+1) &= w(x_i, y_i, 1)x_i^t + w(y_i, y_i, 1)y_i^t + w(y_i, y_i, 2)y_i^{t-1} + w(a_s, y_i, 1)a_s^t + w(a_1, y_i, 1)a_1^t \\ &\quad + \sum_{j=2}^{\lceil \log_2 n \rceil} w(a_j, y_i, 1)a_j^t - b(y_i) \\ &\leq 6\gamma + 2\gamma - \gamma - 7\gamma/2 - \ln 2 - (l-1) \cdot \ln 2 - 11\gamma/2 \\ &\leq -2\gamma.\end{aligned}$$

Thus, $\mathbb{P}[y_i^{t+1} = 1 | N^t = C, N^{t-1} = C'] \leq e^{-2\gamma}$. By a union bound over at most n such outputs, we have, with probability $\geq 1 - ne^{-2\gamma}$, $y_i^{t+1} \leq \min(y_i^t, y_i^{t-1})$ for all i, giving conclusion (1) and completing the lemma. □

We now formalize the fact that the network converges to a valid WTA output configuration in just a single step with constant probability, as long as the number of inhibitors matches the minimum number of firing outputs in the preceding two steps. Corollary 5.5.13 can be viewed as an analog to Lemma 5.3.23 for our two inhibitor networks, except that the number of outputs is reduced to 1, rather than just cut in half, with constant probability.

Corollary 5.5.13 (Constant Probability of a Valid WTA Configuration)**.** *Assume the input execution α_X of $\mathcal{L}_{n,\gamma}$ has X^t fixed for all t and consider configurations C, C' with $C(X) = C'(X) = X^t$, $C(a_s) = C(a_1) = ...C(a_l) = 1$, $C(a_{l+1}) = ... = C(a_{\lceil \log_2 n \rceil}) = 0$ for some $l \geq 1$, and $C(y_i) \leq C(x_i)$, $C'(y_i) \leq C'(x_i)$ for all i. For any time $t \geq 1$, if $\| \min(C', C)\|_1 \in \left[2^l, 2^{l+1}\right)$, then*

$$\mathbb{P}[Y^{t+1} \textit{ is a valid WTA output configuration } | N^t = C, N^{t-1} = C'] \geq \frac{1}{16} - ne^{-2\gamma}.$$

Proof. Let $\bar{Y}$ be the set of outputs who fire in both in C, C'. So $\|\bar{Y}\|_1 = \| \min(C', C)\|_1$. By conclusions (2) and (3) of Lemma 5.5.12, and the assumption that $\| \min(C', C)\|_1 \in \left[2^l, 2^{l+1}\right)$:

$$\begin{aligned}
\mathbb{P}[\|\bar{Y}^{t+1}\|_1 = 1 | N^t = C, N^{t-1} = C'] &= \frac{1}{1+2^l} \cdot \left(1 - \frac{1}{1+2^l}\right)^{\|\bar{Y}^t\|_1} \cdot \|\bar{Y}^t\|_1 \\
&\geq \frac{1}{1+2^l} \left(1 - \frac{1}{1+2^l}\right)^{2^{l+1}} \cdot 2^l \\
&\geq \frac{2^l}{1+2^l} \cdot \frac{1}{8} \geq \frac{1}{16}.
\end{aligned}$$

Further, by conclusion (1) of Lemma 5.5.12, no output outside if $\bar{Y}$ fires at time $t+1$ with probability $\geq 1 - ne^{-2\gamma}$. Additionally, by assumption, for all $y_i \in \bar{Y}$, $x_i^t = 1$. So since if exactly one output in $\bar{Y}$ fires, N^{t+1} is a valid WTA output configuration. This gives the corollary by a union bound. □

We also show a related corollary – if the number of firing inhibitors *exceeds* the appropriate amount for the number of firing outputs, then with good probability, no outputs fire in the next time step.

Corollary 5.5.14 (Constant Probability of Zero Firing Outputs)**.** *Assume the input execution α_X of $\mathcal{L}_{n,\gamma}$ has X^t fixed for all t and consider configurations C, C' with $C(X) = C'(X) = X^t$, $C(a_s) = C(a_1) = ...C(a_l) = 1$, $C(a_{l+1}) = ... = C(a_{\lceil \log_2 n \rceil}) = 0$ for some $l \geq 1$, and $C(y_i) \leq C(x_i)$, $C'(y_i) \leq C'(x_i)$ for all i. For any time $t \geq 1$, if $\| \min(C', C)\|_1 \in \left[0, 2^{l+1}\right)$, then*

$$\mathbb{P}[\|Y^{t+1}\|_1 = 0 | N^t = C, N^{t-1} = C'] \geq \frac{1}{8} - ne^{-2\gamma}$$

Proof. Let $\bar{Y}$ be the set of outputs who fire in both in C, C'. So $\|\bar{Y}\|_1 = \| \min(C', C)\|_1 \in \left[0, 2^{l+1}\right)$. By conclusions (2) and (3) of Lemma 5.5.12:

$$\begin{aligned}\mathbb{P}[\|\bar{Y}^{t+1}\|_1 = 0 | N^t = C, N^{t-1} = C'] &= \left(1 - \frac{1}{1 + 2^l}\right)^{\|\bar{Y}^t\|_1} \\ &\geq \left(1 - \frac{1}{1 + 2^l}\right)^{2^{l+1}} \\ &\geq \frac{1}{8}.\end{aligned}$$

Further, by conclusion (1) of Lemma 5.5.12, no output outside if $\bar{Y}$ fires at time $t+1$ with probability $\geq 1 - ne^{-2\gamma}$. This gives the corollary by a union bound. □

Finally, we show that if there is no inhibition in the network, all outputs corresponding to firing inputs are likely to fire at the next time step.

Lemma 5.5.15 (No Inhibitor Effect)**.** *For any time $t \geq 1$ and configurations C, C' of $\mathcal{L}_{n,\gamma}$, if $\|C(A)\|_1 = 0$, then*

$$\mathbb{P}[y_i^{t+1} = x_i^t \text{ for all } i | N^t = C, N^{t-1} = C'] \geq 1 - ne^{-\gamma/2}.$$

Proof. We consider two cases:

Case 1: $x_i^t = 0$.

In this case:

$$\begin{aligned}\mathrm{pot}(y_i,t+1) &= w(x_i,y_i,1)x_i^t + w(y_i,y_i,1)y_i^t + w(y_i,y_i,2)y_i^{t-1} + w(a_s,y_i,1)a_s^t + \\ &\quad + \sum_{j=1}^{\lceil \log_2 n\rceil} w(a_j,y_i,1)a_j^t - b(y_i)\\ &\leq 0 + 4\gamma + 0 + 0 - 11\gamma/2\\ &\leq -3\gamma/2.\end{aligned}$$

This gives

$$\mathbb{P}[y_i^{t+1} = 0 = x_i^t | N^t = C, N^{t-1} = C'] \geq 1 - e^{-2\gamma}.$$

Case 2: $x_i^t = 1$.

In this case:

$$\begin{aligned}\mathrm{pot}(y_i,t+1) &= w(x_i,y_i,1)x_i^t + w(y_i,y_i,1)y_i^t + w(y_i,y_i,2)y_i^{t-1} + w(a_s,y_i,1)a_s^t + \\ &\quad + \sum_{j=1}^{\lceil \log_2 n\rceil} w(a_j,y_i,1)a_j^t - b(y_i)\\ &\geq 6\gamma + 0 + 0 + 0 - 11\gamma/2\\ &\geq \gamma/2.\end{aligned}$$

This gives

$$\mathbb{P}[y_i^{t+1} = 1 = x_i^t | N^t = C, N^{t-1} = C'] \geq 1 - e^{-\gamma/2}.$$

The lemma then follows after union bounding over all n outputs. □

5.5.4 Stability

In this section we show that once in a valid WTA output configuration (Definition 5.2.6), the network remains in this configuration with high probability. Due to our use of a length-two history period, our stability proof requires certain conditions on the firing states at both times t and times $t-1$. We will focus on the case when there is at least one firing input (i.e., when $\|X^t\| \geq 1$.) In the case $\|X^t\| = 0$, convergence to a valid WTA output configuration and stability of this configuration follow easily from Lemma 5.5.5.

Definition 5.5.16 below can be viewed as a two-step generalization of a near-valid WTA configuration of our two-inhibitor networks (Definition 5.3.15).

Definition 5.5.16 (Near-Stable Pair of Configurations)**.** *Assume the input execution α_X of $\mathcal{L}_{n,\gamma}$ has X^t fixed for all t and $\|X^t\|_1 \geq 1$. Consider configurations C, C' with $C(X) = C'(X) = X^t$. The ordered pair (C', C) is* near-stable *if:*

1. *$\|\max(C'(Y), C(Y))\|_1 = 1$, where $\max(C'(Y), C(Y))$ is the entrywise maximum of $C'(Y), C(Y)$. This condition requires that exactly one output fires in configurations C', C. It may fire in one or both configurations.*

2. *$C(a_s) = C'(a_s) = 1$.*

3. *$C(a_i) = 0$ for all $i \in \{1, \lceil \log_2 n \rceil\}$.*

4. *$C(y_i) \leq C(x_i)$, $C'(y_i) \leq C'(x_i)$ for all i.*

Note that by conditions (1) and (4), at least one of $C(Y), C'(Y)$ is a valid WTA output configuration (Definition 5.2.6). In our proofs, it will be useful to refer to the output whose existence is guaranteed by condition (1). Thus we define:

Definition 5.5.17. *Assume the input execution α_X of $\mathcal{L}_{n,\gamma}$ has X^t fixed for all t and $\|X^t\|_1 \geq 1$. For any near-stable pair of configurations (C', C) with $C(X) = C'(X) = X^t$, let $out(C', C) \in \{1, ..., n\}$ be equal to the index of the unique output that fires in C', C (whose existence is guaranteed by condition (1) of Definition 5.5.16).*

We next show that if the configurations (N^{t-1}, N^t) are near-stable, then with high probability, N^{t+1} will be a valid WTA output configuration. Further, the network will stabilize for t_s steps. That is, with high probability, we will have $N^{t+1} = ... = N^{t+t_s+1}$. Lemma 5.5.16 is analogous to Lemma 5.3.21 for our two-inhibitor networks.

Lemma 5.5.18 (Reaching Stability From Near-Stable Configurations)**.** *Assume the input execution α_X of $\mathcal{L}_{n,\gamma}$ has X^t fixed for all t and that $\|X^t\|_1 \geq 1$. Consider any near-stable pair of configurations (C', C) with $C(X) = C'(X) = X^t$. For any time $t \geq 1$, conditioned on $N^t = C, N^{t-1} = C'$, with probability $\geq 1 - (n + \lceil \log_2 n \rceil + 1) \cdot e^{-\gamma/2}$,*

1. *N^{t+1} is a valid WTA output configuration (Definition 5.2.6).*

2. *$y^{t+1}_{out(N^{t-1}, N^t)} = 1$. That is, the winner at time $t+1$ is the output firing in N^{t-1}, and/or N^t.*

3. *(N^t, N^{t+1}) is also a near-stable pair of configurations.*

Proof. By condition (1) of Definition 5.5.16, for all $j \neq out(N^{t-1}, N^t)$, $y_j^t = y_j^{t-1} = 0$. Additionally, by conditions (2) and (3), a_s is the only inhibitor that fires at time t. So by Lemma 5.5.11,

$$\mathbb{P}[Y^{t+1} = \max(Y^t, Y^{t-1}) | N^t = C] \geq 1 - ne^{-\gamma/2}. \tag{5.68}$$

This gives that $y_{out(N^{t-1},N^t)}^{t+1} = 1$ while $y_j^{t+1} = 0$ for all $j \neq out(N^{t-1}, N^t)$. By condition (4) of Definition 5.5.16 we must have also have $x_{out(N^{t-1},N^t)}^{t+1} = 1$. This implies conclusions (1) and (2) of the lemma. It remains to show conclusion (3).

Condition (1) of Definition 5.5.16 holds if $Y^{t+1} = \max(Y^t, Y^{t-1})$ (see (5.68)) since $y_{out(N^{t-1},N^t)}^{t+1} = 1$ and further, $y_{out(N^{t-1},N^t)}$ is the only output that may fire at time t. Conditions (2) and (3) with probability $\geq 1 - (\lceil \log_2 n \rceil + 1) \cdot e^{-\gamma/2}$ conditioned on $N^t = C, N^{t-1} = C'$ by Lemmas 5.5.7 and 5.5.8 and a union bound. Finally, condition (4) holds if $Y^{t+1} = \max(Y^t, Y^{t-1})$ (see (5.68)). Overall, by a union bound, all three conclusions hold with probability $\geq 1 - (n + \lceil \log_2 n \rceil + 1) \cdot e^{-\gamma/2}$, giving the lemma.

□

We can use Lemma 5.5.18 to show that $\mathcal{L}_{n,\gamma}$ remains in a valid WTA configuration for t_s consecutive time steps with good probability.

Corollary 5.5.19 (Stability of Valid WTA Configurations)**.** *Assume the input execution α_X of $\mathcal{L}_{n,\gamma}$ has X^t fixed for all t and that $\|X^t\|_1 \geq 1$. Consider any near-stable pair of configurations (C', C) with $C(X) = C'(X) = X^t$. For any time $t \geq 1$, . For any time t conditioned on $N^t = C, N^{t-1} = C'$, with probability $\geq 1 - 3t_s n \cdot e^{-\gamma/2}$, Y^{t+1} is a valid WTA output configuration and further,*

$$Y^{t+1} = Y^{t+2} = ... = Y^{t+t_s+1}.$$

Proof. We apply Lemma 5.5.18 for each time $t+1, ..., t+t_s+1$ in succession. This is possible since by conclusion (3), if (N^{t-1}, N^{t-1}) is a near-stable pair of configurations, then with high probability (N^t, N^{t+1}) is as well. Conclusion (1) gives that $N^{t+1}, ..., N^{t+t_s+1}$ are all valid WTA configurations and conclusion (2) gives that $Y^{t+1} = Y^{t+2} = ... = Y^{t+t_s+1}$. By a union bound over these t_s steps, the conclusion holds with probability $\geq 1 - t_s(n + \lceil \log_2 n \rceil + 1) \cdot e^{-\gamma/2} \geq 1 - 3t_s n \cdot e^{-\gamma/2}$. □

5.5.5 Convergence in $O(1)$ Steps

We now use the transition lemmas of Section 5.5.3 to show that, starting from any configuration, the network converges to a near-stable pair of configurations (Definition 5.5.16) with constant probability in $O(1)$ time steps. Combined with Corollary 5.5.19 this shows convergence to a valid WTA configuration (and stability within this configuration for t_s steps) in $O(1)$ steps with constant probability.

Our analysis is tedious by straightforward. It breaks down into nine cases, based on the initial output and inhibitor behavior. These cases are summarized in Table 5.2. Since some cases depend on our bounds for others, we do not prove them in the order listed.

Output Count $\|Y^t\|_1$	Inhibitor Count $\|A^t\|_1$	Lemma
0	0	Lemma 5.5.20
0	1	Lemma 5.5.27
0	any $a > 1$	Lemma 5.5.28
1	0	Lemma 5.5.24
1	1	Lemma 5.5.25
1	any $a > 1$	Lemma 5.5.26
any $k > 1$	0	Lemma 5.5.23
any $k > 1$	1	Lemma 5.5.22
any $k > 1$	any $a > 1$	Lemma 5.5.21

Table 5.2: Summary of cases from which we show convergence in $O(1)$ steps to a near-stable pair of configurations (Definition 5.5.16) with constant probability.

Lemma 5.5.20 ($\|Y^t\|_1 = \|A^t\|_1 = 0$)**.** *Assume the input execution α_X of $\mathcal{L}_{n,\gamma}$ has X^t fixed for all t and that $\|X^t\|_1 \geq 1$. Consider any pair of configurations C', C with $C(X) = C'(X) = X^t$ and $\|C(Y)\|_1 = \|C(A)\|_1 = 0$. For any time $t \geq 1$,*

$$\mathbb{P}[(N^{t+3}, N^{t+4}) \text{ is near-stable } | N^t = C, N^{t-1} = C'] \geq \frac{1}{16} - 12n \cdot e^{-\gamma/2}.$$

Proof. The proof follows from a series of four steps, arguing about the state of $\mathcal{L}_{n,\gamma}$ at times $t+1, t+2, t+3, t+4$.

Step 1:

Let $\mathcal{E}_1$ be the event that $y_i^{t+1} = x_i^{t+1}$ for all i and that $a_i^{t+1} = 0$ for all $i \in \{1, ..., \lceil \log_2 n \rceil\}$. By Lemma 5.5.15, since $\|C(A)\|_1 = 0$,

$$\mathbb{P}[y_i^{t+1} = x_i^{t+1} \text{ for all } i | N^t = C, N^{t-1} = C'] \geq 1 - ne^{-\gamma/2}.$$

Additionally, since $\|C(Y)\|_1 = 0$, by Lemma 5.5.8 conclusion (1),

$$\mathbb{P}[a_i^{t+1} = 0 \text{ for all } i \in \{1, ..., \lceil \log_2 n \rceil\} | N^t = C, N^{t-1} = C'] \geq 1 - \lceil \log_2 n \rceil \cdot e^{-\gamma/2}.$$

Thus, by a union bound we have:

$$\mathbb{P}[\mathcal{E}_1 | N^t = C, N^{t-1} = C'] \geq 1 - (n + \lceil \log_2 n \rceil) \cdot e^{-\gamma/2}. \tag{5.69}$$

Step 2:

Let $\mathcal{E}_2$ be the event that $y_i^{t+2} = x_i^{t+2}$ for all i and that for $l = \lfloor \log_2 (\|X^t\|_1) \rfloor$, $a_s^{t+2} = a_1^{t+2} = ... = a_l^{t+2} = 1$ and $a_{l+1}^{t+2} = ... = a_{\lceil \log_2 n \rceil}^{l+2} = 0$ (if $l = 0$, just $a_s^{t+2} = 1$). Conditioned on $\mathcal{E}_1$, the only inhibitor that possibly fires at time $t+1$ is a_s. We can separately consider the cases when $a_s^{t+1} = 0$ and when $a_s^{t+1} = 1$. By Lemma 5.5.11,

$$\mathbb{P}\left[y_i^{t+2} = x_i^{t+2} \text{ for all } i | \mathcal{E}_1, a_s^{t+1} = 1, N^t = C, N^{t-1} = C'\right] \geq 1 - ne^{-\gamma/2}.$$

By Lemma 5.5.15 we also have

$$\mathbb{P}\left[y_i^{t+2} = x_i^{t+2} \text{ for all } i | \mathcal{E}_1, a_s^{t+1} = 0, N^t = C, N^{t-1} = C'\right] \geq 1 - ne^{-\gamma/2}.$$

By the law of total probability this gives:

$$\mathbb{P}\left[y_i^{t+2} = x_i^{t+2} \text{ for all } i | \mathcal{E}_1, N^t = C, N^{t-1} = C'\right] \geq 1 - ne^{-\gamma/2}. \tag{5.70}$$

We also apply Lemma 5.5.8. Conditioned on $\mathcal{E}_1$, for $l = \lfloor \log_2 (\|X^t\|_1) \rfloor$, we have $\|Y^{t+1}\|_1 = \|X^t\|_1 \in [2^l, 2^{l+1})$, which gives that,

$$\mathbb{P}[a_1^{t+2} = ... = a_i^{t+2} = 1 \text{ and } a_{i+1}^{t+2} = ... = a_{\lceil \log_2 n \rceil}^{t+2} = 0 | \mathcal{E}_1, N^t = C, N^{t-1} = C'] \\ \geq 1 - \lceil \log_2 n \rceil \cdot e^{-\gamma/2}. \tag{5.71}$$

Similarly, applying Lemma 5.5.7, since conditioned on $\mathcal{E}_1$, $\|Y^{t+1}\|_1 = \|X^{t+1}\|_1 \geq 1$:

$$\mathbb{P}[a_s^{t+2} = 1|\mathcal{E}_1, N^t = C, N^{t-1} = C'] \geq 1 - e^{-\gamma/2}. \tag{5.72}$$

Combining (5.70), (5.71), and (5.72) we have:

$$\mathbb{P}[\mathcal{E}_2|\mathcal{E}_1, N^t = C, N^{t-1} = C'] \geq 1 - (n + \lceil \log_2 n \rceil + 1) \cdot e^{-\gamma/2}. \tag{5.73}$$

Step 3:

Let $\mathcal{E}_3$ be the event that Y^{t+3} is a valid WTA configuration, and that for $l = \lfloor \log_2 (\|X^t\|_1) \rfloor$, $a_s^{t+3} = a_1^{t+3} = ... = a_l^{t+3} = 1$ and $a_{l+1}^{t+3} = ... = a_{\lceil \log_2 n \rceil}^{l+3} = 0$.

If $l = 0$, conditioned on $\mathcal{E}_1, \mathcal{E}_2$, we have $Y^{t+1} = Y^{t+2} = X^t$ and so $\|Y^{t+1}\|_1 = \|Y^{t+2}\|_1 = \|X^t\|_1 = 1$. By the stability property of Lemma 5.5.11 we thus have:

$$\mathbb{P}[Y^{t+3} \text{ is a valid WTA output configuration } |\mathcal{E}_1, \mathcal{E}_2, N^t = C, N^{t-1} = C'] \geq 1 - ne^{-\gamma/2}.$$

Ofthterwise, for $l \geq 1$, by Corollary 5.5.13, since conditioned on $\mathcal{E}_1$ and $\mathcal{E}_2$,

$$\| \min(Y^{t+1}, Y^{t+2})\|_1 = \|X^t\|_1 \in [2^l, 2^{l+1})$$

and $a_s^{t+2} = a_1^{t+2} = ... = a_l^{t+2} = 1$ and $a_{l+1}^{t+2} = ... = a_{\lceil \log_2 n \rceil}^{t+2} = 0$,

$$\mathbb{P}[Y^{t+3} \text{ is a valid WTA output configuration } |\mathcal{E}_1, \mathcal{E}_2, N^t = C, N^{t-1} = C'] \geq \frac{1}{16} - ne^{-2\gamma}.$$

We can easily bound the probability of $a_s^{t+3} = a_1^{t+3} = ... = a_l^{t+3} = 1$ and $a_{l+1}^{t+3} = ... = a_{\lceil \log_2 n \rceil}^{l+3} = 0$ using the same arguments as in (5.71) and (5.72), giving, via a union bound:

$$\mathbb{P}[\mathcal{E}_3|\mathcal{E}_1, \mathcal{E}_2, N^t = C, N^{t-1} = C'] \geq \frac{1}{16} - (n + \lceil \log_2 n \rceil + 1) \cdot e^{-\gamma/2}. \tag{5.74}$$

Step 4:

Finally, let $\mathcal{E}_4$ be the event that $\max(Y^{t+3}, Y^{t+4}) = 1$, $a_s^{t+4} = 1$, $\sum_{j=1}^{\lceil \log_2 n \rceil} a_j^{t+4} = 0$ and $y_i^{t+4} \leq x_i^{t+4}$ for all i. We can check via Definition 5.5.16 that if $\mathcal{E}_3$ and $\mathcal{E}_4$ occur, then (N^{t+3}, N^{t+4}) is a near-stable pair.

Since conditioned on $\mathcal{E}_3$, $a_s^{t+3} = a_1^{t+3} = ... = a_l^{t+3} = 1$ and $a_{l+1}^{t+3} = ... = a_{\lceil \log_2 n \rceil}^{l+3} = 0$, if $l \geq 1$, by Lemma 5.5.12 conclusion (1),

$$\mathbb{P}[\|Y^{t+4}\|_1 \leq \|Y^{t+3}\|_1 | \mathcal{E}_1, \mathcal{E}_2, \mathcal{E}_3, N^t = C, N^{t-1} = C'] \geq 1 - ne^{-\gamma/2}. \tag{5.75}$$

Since conditioned on $\mathcal{E}_3$, $\|Y^{t+3}\|_1 = 1$, this gives $\max(Y^{t+3}, Y^{t+4}) = 1$. If $l = 0$, then we have an identical bound via the stability property of Lemma 5.5.11.

Again, since conditioned on $\mathcal{E}_3$, $\|Y^{t+3}\|_1 = 1$, by Lemma 5.5.7,

$$\mathbb{P}[a_s^{t+4} = 1 | \mathcal{E}_1, \mathcal{E}_2, \mathcal{E}_3, N^t = C, N^{t-1} = C'] \geq 1 - e^{-\gamma/2}. \tag{5.76}$$

By Lemma 5.5.8, this also gives

$$\mathbb{P}\left[\sum_{j=1}^{\lceil \log_2 n \rceil} a_j^{t+4} = 0 \big| \mathcal{E}_1, \mathcal{E}_2, \mathcal{E}_3, N^t = C, N^{t-1} = C'\right] \geq 1 - \lceil \log_2 n \rceil \cdot e^{-\gamma/2}. \tag{5.77}$$

By a union bound using (5.75),(5.76), and (5.77),

$$\mathbb{P}[\mathcal{E}_4 | \mathcal{E}_1, \mathcal{E}_2, \mathcal{E}_3, N^t = C, N^{t-1} = C'] \geq 1 - (n + \lceil \log_2 n \rceil + 1) \cdot e^{-\gamma/2}. \tag{5.78}$$

Completing the proof:

Let $\mathcal{E}$ be the event that (N^{t+3}, N^{t+4}) is near-stable . We can complete the proof by bounding:

$$\begin{aligned}
\mathbb{P}[\mathcal{E} | N^t = C, N^{t-1} = C'] &\geq \mathbb{P}[\mathcal{E}_3, \mathcal{E}_4 | N^t = C, N^{t-1} = C'] \\
&\geq \mathbb{P}[\mathcal{E}_4 | \mathcal{E}_1, \mathcal{E}_2, \mathcal{E}_3, N^t = C, N^{t-1} = C'] \\
&\quad \cdot \mathbb{P}[\mathcal{E}_3 | \mathcal{E}_1, \mathcal{E}_2, N^t = C, N^{t-1} = C'] \\
&\quad \cdot \mathbb{P}[\mathcal{E}_2 | \mathcal{E}_1, N^t = C, N^{t-1} = C'] \\
&\quad \cdot \mathbb{P}[\mathcal{E}_1 | N^t = C, N^{t-1} = C'].
\end{aligned}$$

We can bound the above terms using (5.69), (5.73),(5.74), and (5.78) giving:

$$\begin{aligned}
\mathbb{P}[\mathcal{E} | N^t = C, N^{t-1} = C'] &\geq \left(1 - (n + \lceil \log_2 n \rceil) \cdot e^{-\gamma/2}\right) \cdot \left(1 - (n + \lceil \log_2 n \rceil + 1) \cdot e^{-\gamma/2}\right) \\
&\quad \cdot \left(\frac{1}{16} - (n + \lceil \log_2 n \rceil + 1) \cdot e^{-\gamma/2}\right) \cdot \left(1 - (n + \lceil \log_2 n \rceil + 1) \cdot e^{-\gamma/2}\right) \\
&\geq \frac{1}{16} - 4(n + \lceil \log_2 n \rceil + 1) \cdot e^{-\gamma/2} \\
&\geq \frac{1}{16} - 12n \cdot e^{-\gamma/2}.
\end{aligned}$$

□

Lemma 5.5.21 ($\|Y^t\|_1 > 1$, $\|A^t\|_1 > 1$)**.** *Assume the input execution α_X of $\mathcal{L}_{n,\gamma}$ has X^t fixed for all t and that $\|X^t\|_1 \geq 1$. Consider any pair of configurations C', C with $C(X) = C'(X) = X^t$, $\|C(Y)\|_1 > 1$, $\|C(A)\|_1 > 1$, $C(a_s) = 1$, and $C(y_i) \leq C(x_i)$, $C'(y_i) \leq C'(x_i)$ for all i. For any time $t \geq 1$,*

$$\mathbb{P}[(N^{t+i}, N^{t+i+1}) \text{ is near-stable for some } i \leq 7 | N^t = C, N^{t-1} = C'] \geq \frac{1}{128} - 24n \cdot e^{-\gamma/2}.$$

Proof. Again the proof follows from a series of steps, arguing about the state of $\mathcal{L}_{n,\gamma}$ at times $t+1, t+2, \ldots$. Let $\mathcal{E}$ be the event that (N^{t+i}, N^{t+i+1}) is near-stable for some $i \leq 7$.

Step 1:

Let $\mathcal{E}_1$ be the event that $y_i^{t+1} \leq y_i^t$ for all i and that for $l = \lfloor \log_2 (\|Y^t\|_1) \rfloor$, $a_s^{t+2} = a_1^{t+2} = \ldots = a_l^{t+2} = 1$ and $a_{l+1}^{t+2} = \ldots = a_{\lceil \log_2 n \rceil}^{l+2} = 0$ (note that $l \geq 1$ since $\|Y^t\|_1 > 1$). Let $\mathcal{E}_{1,0}$ be the event that $\mathcal{E}_1$ holds and $\|Y^{t+1}\|_1 = 0$. Let $\mathcal{E}_{1,1}$ be the event that $\mathcal{E}_1$ holds and $\|Y^{t+1}\|_1 = 1$. Finally, let $\mathcal{E}_{1,>1}$ be the event that $\mathcal{E}_1$ holds and $\|Y^{t+1}\|_1 > 1$. Applying Lemmas 5.5.7, 5.5.8, and 5.5.12 and a union bound:

$$\mathbb{P}[\mathcal{E}_1 | N^t = C, N^{t-1} = C'] \geq 1 - (n + \lceil \log_2 n \rceil + 1) \cdot e^{-\gamma/2}. \tag{5.79}$$

Step 2:

Let $\mathcal{E}_2$ be the event that $\|Y^{t+2}\|_1 = 0$, that $a_s^{t+2} = 1$ if $\|Y^{t+1}\|_1 > 0$, and that for $l' = \lfloor \log_2 (\|Y^{t+1}\|_1) \rfloor$, $a_1^{t+2} = \ldots = a_{l'}^{t+2} = 1$ and $a_{l'+1}^{t+2} = \ldots = a_{\lceil \log_2 n \rceil}^{l+2} = 0$. Since, conditioned on $\mathcal{E}_1$, $\|Y^{t+1}\|_1 \leq \|Y^t\|_1 \leq 2^{l+1}$, $a_s^{t+2} = a_1^{t+2} = \ldots = a_l^{t+2} = 1$ and $a_{l+1}^{t+2} = \ldots = a_{\lceil \log_2 n \rceil}^{l+2} = 0$, by Corollary 5.5.14 combined with Lemmas 5.5.7, 5.5.8:

$$\mathbb{P}[\mathcal{E}_2 | \mathcal{E}_1, N^t = C, N^{t-1} = C'] \geq \frac{1}{8} - (n + \lceil \log_2 n \rceil + 1) \cdot e^{-\gamma/2}. \tag{5.80}$$

We now write:

$$\begin{aligned}\mathbb{P}[\mathcal{E} | N^t = C, N^{t-1} = C'] \geq \min \big(&\mathbb{P}[\mathcal{E} | \mathcal{E}_2, \mathcal{E}_{1,0}, N^t = C, N^{t-1} = C'], \\ &\mathbb{P}[\mathcal{E} | \mathcal{E}_2, \mathcal{E}_{1,1}, N^t = C, N^{t-1} = C'], \\ &\mathbb{P}[\mathcal{E} | \mathcal{E}_2, \mathcal{E}_{1,>1}, N^t = C, N^{t-1} = C'] \big) \\ &\cdot \mathbb{P}[\mathcal{E}_2 | \mathcal{E}_1, N^t = C, N^{t-1} = C'] \cdot \mathbb{P}[\mathcal{E}_1 | N^t = C, N^{t-1} = C']\end{aligned}$$

Using (5.79) and (5.80) we can bound the above by:

$$\begin{aligned}\mathbb{P}[\mathcal{E}|N^t = C, N^{t-1} = C'] \geq \min\big(&\mathbb{P}[\mathcal{E}|\mathcal{E}_2, \mathcal{E}_{1,0}, N^t = C, N^{t-1} = C'],\\ &\mathbb{P}[\mathcal{E}|\mathcal{E}_2, \mathcal{E}_{1,1}, N^t = C, N^{t-1} = C'],\\ &\mathbb{P}[\mathcal{E}|\mathcal{E}_2, \mathcal{E}_{1,>1}, N^t = C, N^{t-1} = C']\big) \cdot \left(\frac{1}{8} - 6n \cdot e^{-\gamma/2}\right).\end{aligned} \tag{5.81}$$

We bound the minimum above by considering each of the three cases separately.

Case 1: $\mathbb{P}[\mathcal{E}|\mathcal{E}_2, \mathcal{E}_{1,0}, N^t = C, N^{t-1} = C']$**.**

Conditioned on $\mathcal{E}_2$, $\|Y^{t+2}\|_1 = 0$. So by Lemma 5.5.20,

$$\mathbb{P}[(N^{t+5}, N^{t+6}) \text{ is near-stable } |\mathcal{E}_2, \mathcal{E}_{1,0}, \|A^{t+2}\|_1 = 0, N^t = C, N^{t-1} = C'] \geq \frac{1}{16} - 12n \cdot e^{-\gamma/2}. \tag{5.82}$$

If $\|A^{t+2}\| \geq 1$ then, conditioned on $\mathcal{E}_2$, we must have $a_s^{t+2} = 1$. Let $\mathcal{E}_3$ be the event that $\|Y^{t+3}\|_1 = \|A^{t+3}\|_1 = 0$. By Lemmas 5.5.7, 5.5.8, and 5.5.11 and the fact that conditioned on $\mathcal{E}_{1,0}$, $\|Y^{t+1}\|_1 = 0$,

$$\mathbb{P}[\mathcal{E}_3|\mathcal{E}_2, \mathcal{E}_{1,0}, \|A^{t+2}\|_1 \geq 1, N^t = C, N^{t-1} = C'] \geq 1 - (n + \lceil \log_2 n \rceil + 1) \cdot e^{-\gamma/2} \tag{5.83}$$

Again by Lemma 5.5.20,

$$\mathbb{P}[(N^{t+6}, N^{t+7}) \text{ is near-stable } |\mathcal{E}_3, \mathcal{E}_2, \mathcal{E}_{1,0}, \|A^{t+2}\|_1 = 0, N^t = C, N^{t-1} = C'] \geq \frac{1}{16} - 12n \cdot e^{-\gamma/2}.$$

Combined with (5.83) this gives

$$\mathbb{P}[(N^{t+6}, N^{t+7}) \text{ is near-stable } |\mathcal{E}_2, \mathcal{E}_{1,0}, \|A^{t+2}\|_1 \geq 1, N^t = C, N^{t-1} = C'] \geq \frac{1}{16} - 15n \cdot e^{-\gamma/2}.$$

By the law of total probability, combined with (5.82) we have

$$\mathbb{P}[\mathcal{E}|\mathcal{E}_2, \mathcal{E}_{1,0}, N^t = C, N^{t-1} = C'] \geq \frac{1}{16} - 15n \cdot e^{-\gamma/2} \tag{5.84}$$

which completes this case.

Case 2: $\mathbb{P}[\mathcal{E}|\mathcal{E}_2, \mathcal{E}_{1,1}, N^t = C, N^{t-1} = C']$**.**

In this case, conditioned on $\mathcal{E}_{1,1}$ and $\mathcal{E}_2$, $a_s^{t+1} = a_s^{t+2} = 1$, a_s is the only inhibitor that fires at time $t+2$, $\|Y^{t+1}\|_1 = 1$, and $\|Y^{t+2}\|_1 = 0$. Thus, (N^{t+1}, N^{t+2}) is a near-stable pair of configurations, and so vacuously,

$$\mathbb{P}[\mathcal{E}|\mathcal{E}_2, \mathcal{E}_{1,0}, N^t = C, N^{t-1} = C'] = 1 \tag{5.85}$$

which completes this case.

Case 3: $\mathbb{P}[\mathcal{E}|\mathcal{E}_2, \mathcal{E}_{1,>1}, N^t = C, N^{t-1} = C']$**.**

In this case, conditioned on $\mathcal{E}_{1,>1}$ and $\mathcal{E}_2$, $\|A^{t+2}\|_1 > 1$ and $a_s^{t+2} = 1$. Let $\mathcal{E}_3$ be the event that $\|Y^{t+3}\|_1 = 0$ and that if $\|A^{t+3}\|_1 \geq 1$, $a_s^{t+3} = 1$. By Lemmas 5.5.7, 5.5.8, and 5.5.12,

$$\mathbb{P}[\mathcal{E}_3|\mathcal{E}_2, \mathcal{E}_{1,>1}, N^t = C, N^{t-1} = C'] \geq 1 - (n + \lceil \log_2 n \rceil + 1)e^{-\gamma/2}. \tag{5.86}$$

In the case that $\|A^{t+3}\|_1 = 0$, we can again apply Lemma 5.5.20 to give:

$$\mathbb{P}[(N^{t+6}, N^{t+7}) \text{ is near-stable } |\mathcal{E}_3, \|A^{t+3}\|_1 = 0, \mathcal{E}_2, \mathcal{E}_{1,>1}, N^t = C, N^{t-1} = C'] \\ \geq \frac{1}{16} - 12n \cdot e^{-\gamma/2}. \tag{5.87}$$

In the case that $\|A^{t+3}\|_1 \geq 1$, let $\mathcal{E}_4$ be the event that $\|Y^{t+4}\|_1 = \|A^{t+4}\| = 0$. We have by Lemmas 5.5.7, 5.5.8 and 5.5.11,

$$\mathbb{P}[\mathcal{E}_4|\mathcal{E}_3, \|A^{t+3}\|_1 \geq 1, \mathcal{E}_2, \mathcal{E}_{1,>1}, N^t = C, N^{t-1} = C'] \geq 1 - (n + \lceil \log_2 n \rceil + 1)e^{-\gamma/2}. \tag{5.88}$$

Further, again by Lemma 5.5.20 we have:

$$\mathbb{P}[(N^{t+7}, N^{t+8}) \text{ is near-stable } |\mathcal{E}_4, \mathcal{E}_3, \|A^{t+3}\|_1 = 0, \mathcal{E}_2, \mathcal{E}_{1,>1}, N^t = C, N^{t-1} = C'] \\ \geq \frac{1}{16} - 12n \cdot e^{-\gamma/2}.$$

Combined with (5.88) this gives:

$$\mathbb{P}[(N^{t+7}, N^{t+8}) \text{ is near-stable } |\mathcal{E}_3, \|A^{t+3}\|_1 = 0, \mathcal{E}_2, \mathcal{E}_{1,>1}, N^t = C, N^{t-1} = C'] \\ \geq \frac{1}{16} - 15n \cdot e^{-\gamma/2}.$$

Further, combined with (5.87), by the law of total probability,

$$\mathbb{P}[\mathcal{E}|\mathcal{E}_3, \mathcal{E}_2, \mathcal{E}_{1,>1}, N^t = C, N^{t-1} = C'] \geq \frac{1}{16} - 15n \cdot e^{-\gamma/2}. \tag{5.89}$$

Finally, combining (5.89) with (5.86) we have:

$$\mathbb{P}[\mathcal{E}|\mathcal{E}_2, \mathcal{E}_{1,>1}, N^t = C, N^{t-1} = C'] \geq \frac{1}{16} - 18n \cdot e^{-\gamma/2} \tag{5.90}$$

completing this case.

Completing the proof.

Using equations (5.84), (5.85), and (5.90) in the three cases above along (5.81):

$$\begin{aligned}\mathbb{P}[\mathcal{E}|N^t = C, N^{t-1} = C'] \geq \min\big(&\mathbb{P}[\mathcal{E}|\mathcal{E}_2, \mathcal{E}_{1,0}, N^t = C, N^{t-1} = C'],\\ &\mathbb{P}[\mathcal{E}|\mathcal{E}_2, \mathcal{E}_{1,1}, N^t = C, N^{t-1} = C'],\\ &\mathbb{P}[\mathcal{E}|\mathcal{E}_2, \mathcal{E}_{1,>1}, N^t = C, N^{t-1} = C']\big) \cdot \left(\frac{1}{8} - 6n \cdot e^{-\gamma/2}\right)\\ \geq &\left(\frac{1}{16} - 18n \cdot e^{-\gamma/2}\right) \cdot \left(\frac{1}{8} - 6n \cdot e^{-\gamma/2}\right)\\ \geq &\frac{1}{128} - 24n \cdot e^{-\gamma/2},\end{aligned}$$

completing the lemma. □

Using Lemma 5.5.21 it is not hard to complete the cases when $\|Y^t\|_1 > 1$ and $\|A^t\|_1 \leq 1$.

Lemma 5.5.22 ($\|Y^t\|_1 > 1$, $\|A^t\|_1 = 1$)**.** *Assume the input execution α_X of $\mathcal{L}_{n,\gamma}$ has X^t fixed for all t and that $\|X^t\|_1 \geq 1$. Consider any pair of configurations C', C with $C(X) = C'(X) = X^t$, $\|C(Y)\|_1 > 1$, $\|C(A)\|_1 = 1$, $C(a_s) = 1$, and $C(y_i) \leq C(x_i)$, $C'(y_i) \leq C'(x_i)$ for all i. For any time $t \geq 1$,*

$$\mathbb{P}[(N^{t+i}, N^{t+i+1}) \textit{ is near-stable for some } i \leq 8 | N^t = C, N^{t-1} = C'] \geq \frac{1}{128} - 28n \cdot e^{-\gamma/2}.$$

Proof. Let $\mathcal{E}$ be the event that (N^{t+i}, N^{t+i+1}) is near-stable for some $i \leq 8$. Let $\mathcal{E}_1$ be the event that $y_i^{t+1} \geq y_i^t$ and $y_i^{t+1} \leq x_i^{t+1}$ for all i, that $a_s^{t+1} = 1$, and that $\|A^{t+1}\|_1 > 1$. By Corollary 5.5.6 and Lemmas 5.5.7, 5.5.8, and 5.5.11, we have:

$$\mathbb{P}[\mathcal{E}_1|N^t = C, N^{t-1} = C'] \geq 1 - (2n + \lceil \log_2 n \rceil + 1)e^{-\gamma/2}.$$

Further, by Lemma 5.5.22 since conditioned on $\mathcal{E}_1$, $\|Y^{t+1}\|_1 > 1$ and $\|A^{t+1}\|_1 > 1$ with $a_s^{t+1} = 1$:

$$\mathbb{P}[\mathcal{E}|\mathcal{E}_1, N^t = C, N^{t-1} = C'] \geq \frac{1}{128} - 24n \cdot e^{-\gamma/2}.$$

This gives the lemma since:

$$\begin{aligned}\mathbb{P}[\mathcal{E}|N^t = C, N^{t-1} = C'] &\geq \mathbb{P}[\mathcal{E}|\mathcal{E}_1, N^t = C, N^{t-1} = C'] \cdot \mathbb{P}[\mathcal{E}_1|N^t = C, N^{t-1} = C'] \\ &\geq \frac{1}{128} - 28n \cdot e^{-\gamma/2}.\end{aligned}$$

□

Lemma 5.5.23 ($\|Y^t\|_1 > 1$, $\|A^t\|_1 = 0$)**.** *Assume the input execution α_X of $\mathcal{L}_{n,\gamma}$ has X^t fixed for all t and that $\|X^t\|_1 \geq 1$. Consider any pair of configurations C', C with $C(X) = C'(X) = X^t$, $\|C(Y)\|_1 > 1$, $\|C(A)\|_1 = 0$, and $C(y_i) \leq C(x_i)$, $C'(y_i) \leq C'(x_i)$ for all i. For any time $t \geq 1$,*

$$\mathbb{P}[(N^{t+i}, N^{t+i+1}) \text{ is near-stable for some } i \leq 8|N^t = C, N^{t-1} = C'] \geq \frac{1}{128} - 27n \cdot e^{-\gamma/2}.$$

Proof. Let $\mathcal{E}$ be the event that (N^{t+i}, N^{t+i+1}) is near-stable for some $i \leq 8$. Let $\mathcal{E}_1$ be the event that $y_i^{t+1} = x_i^{t+1}$ for all i, that $a_s^{t+1} = 1$, and that $\|A^{t+1}\|_1 > 1$. Note that since $\|C(Y)\|_1 > 1$ and $C(y_i) \leq C(x_i)$ for all i, $\mathcal{E}_1$ implies that $\|Y^{t+1}\|_1 = \|X^{t+1}\|_1 > 1$. By Lemmas 5.5.7, 5.5.8, and 5.5.15:

$$\mathbb{P}[\mathcal{E}_1|N^t = C, N^{t-1} = C'] \geq 1 - (n + \lceil \log_2 n \rceil + 1)e^{-\gamma/2}.$$

Further, by Lemma 5.5.22 since conditioned on $\mathcal{E}_1$, $\|Y^{t+1}\|_1 > 1$ and $\|A^{t+1}\|_1 > 1$ with $a_s^{t+1} = 1$:

$$\mathbb{P}[\mathcal{E}|\mathcal{E}_1, N^t = C, N^{t-1} = C'] \geq \frac{1}{128} - 24n \cdot e^{-\gamma/2}.$$

This gives the lemma since:

$$\begin{aligned}\mathbb{P}[\mathcal{E}|N^t = C, N^{t-1} = C'] &\geq \mathbb{P}[\mathcal{E}|\mathcal{E}_1, N^t = C, N^{t-1} = C'] \cdot \mathbb{P}[\mathcal{E}_1|N^t = C, N^{t-1} = C'] \\ &\geq \frac{1}{128} - 27n \cdot e^{-\gamma/2}.\end{aligned}$$

□

We next complete the remaining cases when $\|Y^t\|_1 \leq 1$.

Lemma 5.5.24 ($\|Y^t\|_1 = 1$, $\|A^t\|_1 = 0$)**.** *Assume the input execution α_X of $\mathcal{L}_{n,\gamma}$ has X^t fixed for all t and that $\|X^t\|_1 \geq 1$. Consider any pair of configurations C', C with $C(X) = C'(X) = X^t$, $\|C(Y)\|_1 = 1$, $\|C(A)\|_1 = 0$, and $C(y_i) \leq C(x_i)$, $C'(y_i) \leq C'(x_i)$ for all i. For any time $t \geq 1$,*

$$\mathbb{P}[(N^{t+3}, N^{t+4}) \textit{ is near-stable } | N^t = C, N^{t-1} = C'] \geq \frac{1}{16} - 12n \cdot e^{-\gamma/2}.$$

Proof. The proof follows from a series of four steps, arguing about the state of $\mathcal{L}_{n,\gamma}$ at times $t+1, t+2, t+3, t+4$. The analysis closely mirrors that of Lemma 5.5.20, for the case when $\|Y^t\|_1 = 0$ and $\|A^t\|_1 = 0$.

Step 1:

Let $\mathcal{E}_1$ be the event that $y_i^{t+1} = x_i^{t+1}$ for all i, that $a_i^{t+1} = 0$ for all $i \in \{1, ..., \lceil \log_2 n \rceil\}$, and that $a_s^{t+1} = 1$. By Lemma 5.5.15, since $\|C(A)\|_1 = 0$,

$$\mathbb{P}[y_i^{t+1} = x_i^{t+1} \text{ for all } i | N^t = C, N^{t-1} = C'] \geq 1 - ne^{-\gamma/2}.$$

Additionally, since $\|C(Y)\|_1 = 1$, by Lemma 5.5.8 conclusion (1),

$$\mathbb{P}[a_i^{t+1} = 0 \text{ for all } i \in \{1, ..., \lceil \log_2 n \rceil\} | N^t = C, N^{t-1} = C'] \geq 1 - \lceil \log_2 n \rceil \cdot e^{-\gamma/2}.$$

Finally, by Lemma 5.5.7,

$$\mathbb{P}[a_s^{t+1} = 1 | N^t = C, N^{t-1} = C'] \geq 1 - e^{-\gamma/2}.$$

Thus, by a union bound we have:

$$\mathbb{P}[\mathcal{E}_1 | N^t = C, N^{t-1} = C'] \geq 1 - (n + \lceil \log_2 n \rceil + 1) \cdot e^{-\gamma/2}. \tag{5.91}$$

Step 2:

Let $\mathcal{E}_2$ be the event that $y_i^{t+2} = x_i^{t+2}$ for all i and that for $l = \lfloor \log_2 (\|X^t\|_1) \rfloor$, $a_s^{t+2} = a_1^{t+2} = ... = a_l^{t+2} = 1$ and $a_{l+1}^{t+2} = ... = a_{\lceil \log_2 n \rceil}^{l+2} = 0$ (if $l = 0$, just $a_s^{t+2} = 1$). Conditioned on $\mathcal{E}_1$, the only inhibitor that fires at time $t+1$ is a_s. We can separately consider the cases when $a_s^{t+1} = 0$ and when $a_s^{t+1} = 1$. By Lemma 5.5.11 and the fact that $y_i^{t+1} = x_i^{t+1}$ for all i,

$$\mathbb{P}\left[y_i^{t+2} = x_i^{t+2} \text{ for all } i | \mathcal{E}_1, N^t = C, N^{t-1} = C'\right] \geq 1 - ne^{-\gamma/2}. \tag{5.92}$$

We also apply Lemma 5.5.8. Conditioned on $\mathcal{E}_1$, for $l = \lfloor \log_2(\|X^t\|_1) \rfloor$, we have $\|Y^{t+1}\|_1 = \|X^t\|_1 \in [2^l, 2^{l+1})$, which gives that,

$$\begin{aligned}\mathbb{P}[a_1^{t+2} = ... = a_i^{t+2} = 1 \text{ and } a_{i+1}^{t+2} = ... = a_{\lceil \log_2 n \rceil}^{t+2} = 0 | \mathcal{E}_1, N^t = C, N^{t-1} = C'] \\ \geq 1 - \lceil \log_2 n \rceil \cdot e^{-\gamma/2}.\end{aligned} \tag{5.93}$$

Similarly, applying Lemma 5.5.7, since conditioned on $\mathcal{E}_1$, $\|Y^{t+1}\|_1 = \|X^{t+1}\|_1 \geq 1$:

$$\mathbb{P}[a_s^{t+2} = 1 | \mathcal{E}_1, N^t = C, N^{t-1} = C'] \geq 1 - e^{-\gamma/2}. \tag{5.94}$$

Combining (5.92), (5.93), and (5.94) we have:

$$\mathbb{P}[\mathcal{E}_2 | \mathcal{E}_1, N^t = C, N^{t-1} = C'] \geq 1 - (n + \lceil \log_2 n \rceil + 1) \cdot e^{-\gamma/2}. \tag{5.95}$$

Step 3:

Let $\mathcal{E}_3$ be the event that Y^{t+3} is a valid WTA configuration, and that for $l = \lfloor \log_2(\|X^t\|_1) \rfloor$, $a_s^{t+3} = a_1^{t+3} = ... = a_l^{t+3} = 1$ and $a_{l+1}^{t+3} = ... = a_{\lceil \log_2 n \rceil}^{l+3} = 0$.

If $l = 0$, conditioned on $\mathcal{E}_1, \mathcal{E}_2$, we have $\|Y^{t+1}\|_1 = \|Y^{t+2}\|_1 = 1$ and $Y^{t+1} = Y^{t+2} = X^t$. By the stability property of Lemma 5.5.11 we thus have:

$$\mathbb{P}[Y^{t+3} \text{ is a valid WTA output configuration } | \mathcal{E}_1, \mathcal{E}_2, N^t = C, N^{t-1} = C'] \geq 1 - ne^{-\gamma/2}.$$

Oftherwise, for $l \geq 1$, by Corollary 5.5.13, since conditioned on $\mathcal{E}_1$ and $\mathcal{E}_2$,

$$\|\min(Y^{t+1}, Y^{t+2})\|_1 = \|X^t\|_1 \in [2^l, 2^{l+1})$$

and $a_s^{t+2} = a_1^{t+2} = ... = a_l^{t+2} = 1$ and $a_{l+1}^{t+2} = ... = a_{\lceil \log_2 n \rceil}^{t+2} = 0$,

$$\mathbb{P}[Y^{t+3} \text{ is a valid WTA output configuration } | \mathcal{E}_1, \mathcal{E}_2, N^t = C, N^{t-1} = C'] \geq \frac{1}{16} - ne^{-2\gamma}.$$

We can easily bound the probability of $a_s^{t+3} = a_1^{t+3} = ... = a_l^{t+3} = 1$ and $a_{l+1}^{t+3} = ... = a_{\lceil \log_2 n \rceil}^{l+3} = 0$ using the same arguments as in (5.93) and (5.94), giving, via a union bound:

$$\mathbb{P}[\mathcal{E}_3|\mathcal{E}_1, \mathcal{E}_2, N^t = C, N^{t-1} = C'] \geq \frac{1}{16} - (n + \lceil \log_2 n \rceil + 1) \cdot e^{-\gamma/2}. \tag{5.96}$$

Step 4:

Finally, let $\mathcal{E}_4$ be the event that $\max(Y^{t+3}, Y^{t+4}) = 1$, $a_s^{t+4} = 1$, $\sum_{j=1}^{\lceil \log_2 n \rceil} a_j^{t+4} = 0$ and $y_i^{t+4} \leq x_i^{t+4}$ for all i. We can check via Definition 5.5.16 that if $\mathcal{E}_3$ and $\mathcal{E}_4$ occur, then (N^{t+3}, N^{t+4}) is a near-stable pair.

Since conditioned on $\mathcal{E}_3$, $a_s^{t+3} = a_1^{t+3} = ... = a_l^{t+3} = 1$ and $a_{l+1}^{t+3} = ... = a_{\lceil \log_2 n \rceil}^{l+3} = 0$, if $l \geq 1$, by Lemma 5.5.12 conclusion (1),

$$\mathbb{P}[\|Y^{t+4}\|_1 \leq \|Y^{t+3}\|_1 | \mathcal{E}_1, \mathcal{E}_2, \mathcal{E}_3, N^t = C, N^{t-1} = C'] \geq 1 - ne^{-\gamma/2}. \tag{5.97}$$

Since conditioned on $\mathcal{E}_3$, $\|Y^{t+3}\|_1 = 1$, this gives $\max(Y^{t+3}, Y^{t+4}) = 1$. If $l = 0$, then we have an identical bound via the stability property of Lemma 5.5.11.

Again, since conditioned on $\mathcal{E}_3$, $\|Y^{t+3}\|_1 = 1$, by Lemma 5.5.7,

$$\mathbb{P}[a_s^{t+4} = 1 | \mathcal{E}_1, \mathcal{E}_2, \mathcal{E}_3, N^t = C, N^{t-1} = C'] \geq 1 - e^{-\gamma/2}. \tag{5.98}$$

By Lemma 5.5.8, this also gives

$$\mathbb{P}\left[\sum_{j=1}^{\lceil \log_2 n \rceil} a_j^{t+4} = 0 \middle| \mathcal{E}_1, \mathcal{E}_2, \mathcal{E}_3, N^t = C, N^{t-1} = C'\right] \geq 1 - \lceil \log_2 n \rceil \cdot e^{-\gamma/2}. \tag{5.99}$$

By a union bound using (5.97),(5.98), and (5.99),

$$\mathbb{P}[\mathcal{E}_4|\mathcal{E}_1, \mathcal{E}_2, \mathcal{E}_3, N^t = C, N^{t-1} = C'] \geq 1 - (n + \lceil \log_2 n \rceil + 1) \cdot e^{-\gamma/2}. \tag{5.100}$$

Completing the proof:

Let $\mathcal{E}$ be the event that (N^{t+3}, N^{t+4}) is near-stable . We can complete the proof by bounding:

$$\begin{aligned}\mathbb{P}[\mathcal{E}|N^t = C, N^{t-1} = C'] &\geq \mathbb{P}[\mathcal{E}_3, \mathcal{E}_4|N^t = C, N^{t-1} = C'] \\ &\geq \mathbb{P}[\mathcal{E}_4|\mathcal{E}_1, \mathcal{E}_2, \mathcal{E}_3, N^t = C, N^{t-1} = C'] \\ &\quad \cdot \mathbb{P}[\mathcal{E}_3|\mathcal{E}_1, \mathcal{E}_2, N^t = C, N^{t-1} = C'] \\ &\quad \cdot \mathbb{P}[\mathcal{E}_2|\mathcal{E}_1, N^t = C, N^{t-1} = C'] \\ &\quad \cdot \mathbb{P}[\mathcal{E}_1|N^t = C, N^{t-1} = C']\end{aligned}$$

We can bound the above terms using (5.91), (5.95),(5.96), and (5.100) giving:

$$\begin{aligned}\mathbb{P}[\mathcal{E}|N^t = C, N^{t-1} = C'] \geq &\left(1 - (n + \lceil \log_2 n \rceil + 1) \cdot e^{-\gamma/2}\right) \cdot \left(1 - (n + \lceil \log_2 n \rceil + 1) \cdot e^{-\gamma/2}\right) \\ &\cdot \left(\frac{1}{16} - (n + \lceil \log_2 n \rceil + 1) \cdot e^{-\gamma/2}\right) \cdot \left(1 - (n + \lceil \log_2 n \rceil + 1) \cdot e^{-\gamma/2}\right) \\ &\geq \frac{1}{16} - 4(n + \lceil \log_2 n \rceil + 1) \cdot e^{-\gamma/2} \\ &\geq \frac{1}{16} - 12n \cdot e^{-\gamma/2}.\end{aligned}$$

□

The remaining cases follow relatively straightforwardly from the previous lemmas.

Lemma 5.5.25 ($\|Y^t\|_1 = 1$, $\|A^t\|_1 = 1$)**.** *Assume the input execution α_X of $\mathcal{L}_{n,\gamma}$ has X^t fixed for all t and that $\|X^t\|_1 \geq 1$. Consider any pair of configurations C', C with $C(X) = C'(X) = X^t$, $\|C(Y)\|_1 = 1$, $\|C(A)\|_1 = 1$, $C(a_s) = 1$, and $C(y_i) \leq C(x_i)$, $C'(y_i) \leq C'(x_i)$ for all i. For any time $t \geq 1$,*

$$\mathbb{P}[(N^{t+i}, N^{t+i+1}) \text{ is near-stable for some } i \leq 9 | N^t = C, N^{t-1} = C'] \geq \frac{1}{128} - 28n \cdot e^{-\gamma/2}.$$

Proof. Let $\mathcal{E}$ be the event that (N^{t+i}, N^{t+i+1}) is near-stable for some $i \leq 9$. Let $\mathcal{E}_1$ be the event that $y_i^{t+1} \geq y_i^t$ and $y_i^{t+1} \leq x_i^{t+1}$ for all i, that $a_s^{t+1} = 1$, and that $\|A^{t+1}\|_1 > 1$. By Corollary 5.5.6 and Lemmas 5.5.7, 5.5.8, and 5.5.11, we have:

$$\mathbb{P}[\mathcal{E}_1|N^t = C, N^{t-1} = C'] \geq 1 - (2n + \lceil \log_2 n \rceil + 1)e^{-\gamma/2}. \tag{5.101}$$

We consider two cases, dependent on the number of firing outputs at time $t + 1$. Conditioned on $\mathcal{E}_1$, $\|Y^{t+1}\|_1 \geq \|Y^t\|_1 = 1$. By Lemma 5.5.22 since conditioned on $\mathcal{E}_1$,

$\|A^{t+1}\|_1 > 1$ with $a_s^{t+1} = 1$:

$$\mathbb{P}[\mathcal{E}|\mathcal{E}_1, \|Y^{t+1}\| > 1, N^t = C, N^{t-1} = C'] \geq \frac{1}{128} - 24n \cdot e^{-\gamma/2}.$$

Alternatively, if $\|Y^{t+1}\| = 1$, then (N^t, N^{t+1}) is already a near-stable pair of configurations (Definition 5.5.16) so we vacuously have

$$\mathbb{P}[\mathcal{E}|\mathcal{E}_1, \|Y^{t+1}\| = 1, N^t = C, N^{t-1} = C'] = 1.$$

By the law of total probability this gives:

$$\mathbb{P}[\mathcal{E}|\mathcal{E}_1 N^t = C, N^{t-1} = C'] \geq \frac{1}{128} - 24n \cdot e^{-\gamma/2}.$$

Finally, we obtain the lemma by combining this bound with (5.101). □

Lemma 5.5.26 ($\|Y^t\|_1 = 1$, $\|A^t\|_1 > 1$)**.** *Assume the input execution α_X of $\mathcal{L}_{n,\gamma}$ has X^t fixed for all t and that $\|X^t\|_1 \geq 1$. Consider any pair of configurations C', C with $C(X) = C'(X) = X^t$, $\|C(Y)\|_1 = 1$, $\|C(A)\|_1 > 1$, $C(a_s) = 1$, and $C(y_i) \leq C(x_i)$, $C'(y_i) \leq C'(x_i)$ for all i. For any time $t \geq 1$,*

$$\mathbb{P}[(N^t, N^{t+1}) \text{ is near-stable } | N^t = C, N^{t-1} = C'] \geq 1 - 3n \cdot e^{-\gamma/2}.$$

Proof. Let $\mathcal{E}_1$ be the event that $\|A^{t+1}\|_1 = 1$ and $a_s^{t+1} = 1$. Again Lemmas 5.5.7 and 5.5.8 we have:

$$\mathbb{P}[\mathcal{E}_1 | N^t = C, N^{t-1} = C'] \geq 1 - (\lceil \log_2 n \rceil + 1) \cdot e^{-\gamma/2}.$$

Let $\mathcal{E}_2$ be the event that $\|Y^{t+1}\| \leq 1$ and that $y_i^{t+1} \leq x_i^{t+1}$ for all i. By Lemma 5.5.12,

$$\mathbb{P}[\mathcal{E}_2 | N^t = C, N^{t-1} = C'] \geq 1 - ne^{-\gamma/2}.$$

If $\mathcal{E}_1$ and $\mathcal{E}_2$ both occur, $(N^t, N^t + 1)$ is a near-stable pair of configurations, giving the lemma by a union bound and the fact that $(n + \lceil \log_2 n \rceil + 1) \leq 3n$.

□

Lemma 5.5.27 ($\|Y^t\|_1 = 0$, $\|A^t\|_1 = 1$)**.** *Assume the input execution α_X of $\mathcal{L}_{n,\gamma}$ has X^t fixed for all t and that $\|X^t\|_1 \geq 1$. Consider any pair of configurations C', C with $C(X) = C'(X) = X^t$, $\|C(Y)\|_1 = 0$, $\|C(A)\|_1 = 1$, $C(a_s) = 1$, and $C(y_i) \leq C(x_i)$, $C'(y_i) \leq C'(x_i)$ for all i. For any time $t \geq 1$,*

$$\mathbb{P}[(N^{t+i}, N^{t+i+1}) \textit{ is near-stable for some } i \leq 10 | N^t = C, N^{t-1} = C'] \geq \frac{1}{128} - 31ne^{-\gamma/2}.$$

Proof. Let $\mathcal{E}$ be the event that (N^{t+i}, N^{t+i+1}) is near-stable for some $i \leq 10$. Let $\mathcal{E}_1$ be the event that $a_s^{t+1} = 1$ if $\|Y^{t-1}\|_1 \geq 1$ and 0 otherwise, that $a_i^{t+1} = 0$ for all $i \in \{1, ..., \lceil \log_2 n \rceil\}$, and that $y_i^{t+1} = \max(y_i^{t-1}, y_i^t)$ for all i. By Lemmas 5.5.7, 5.5.8, and 5.5.11

$$\mathbb{P}[\mathcal{E}_1 | N^t = C, N^{t-1} = C'] \geq 1 - (n + \lceil \log_2 n \rceil + 1) \cdot e^{-\gamma/2} \tag{5.102}$$

We next consider two cases, based off the number of firing outputs at time $t - 1$.

Case 1: $\|C'(Y)\|_1 = 0$.

In this case, conditioned on $\mathcal{E}_1$, $\|Y^{t+1}\|_1 = \|A^{t+1}\|_1 = 0$. Thus, by Lemma 5.5.20,

$$\mathbb{P}[(N^{t+4}, N^{t+5}) \text{ is near-stable } | \mathcal{E}_1, N^t = C, N^{t-1} = C'] \geq \frac{1}{16} - 12n \cdot e^{-\gamma/2} \tag{5.103}$$

Case 2: $\|C'(Y)\|_1 \geq 1$.

In this case, conditioned on $\mathcal{E}_1$, $a_s^{t+1} = 1$, $\|A^{t+1}\|_1 = 1$ and $\|Y^{t+1}\|_1 \geq 1$. Thus by Lemmas 5.5.22 and 5.5.25,

$$\mathbb{P}[\mathcal{E} | \mathcal{E}_1, N^t = C, N^{t-1} = C'] \geq \frac{1}{128} - 28n \cdot e^{-\gamma/2} \tag{5.104}$$

Overall by (5.103) and (5.104) we have

$$\mathbb{P}[\mathcal{E} | \mathcal{E}_1, N^t = C, N^{t-1} = C'] \geq \frac{1}{128} - 28n \cdot e^{-\gamma/2},$$

which gives the lemma when combined with (5.102). □

Lemma 5.5.28 ($\|Y^t\|_1 = 0$, $\|A^t\|_1 > 1$)**.** *Assume the input execution α_X of $\mathcal{L}_{n,\gamma}$ has X^t fixed for all t and that $\|X^t\|_1 \geq 1$. Consider any pair of configurations C', C with $C(X) = C'(X) = X^t$, $\|C(Y)\|_1 = 0$, $\|C(A)\|_1 > 1$, $C(a_s) = 1$, and $C(y_i) \leq C(x_i)$, $C'(y_i) \leq C'(x_i)$ for all i. For any time $t \geq 1$,*

$$\mathbb{P}[(N^{t+i}, N^{t+i+1}) \textit{ is near-stable for some } i \leq 11 | N^t = C, N^{t-1} = C'] \geq \frac{1}{128} - 33ne^{-\gamma/2}.$$

Proof. Let $\mathcal{E}$ be the event that (N^{t+i}, N^{t+i+1}) is near-stable for some $i \leq 11$. Let $\mathcal{E}_1$ be the event that $a_i^{t+1} = 0$ for all $i \in \{1, ..., \lceil \log_2 n \rceil\}$, and that $\|Y^{t+1}\|_1 = 0$. By Lemmas 5.5.8 and 5.5.12,

$$\mathbb{P}[\mathcal{E}_1 | N^t = C, N^{t-1} = C'] \geq 1 - (n + \lceil \log_2 n \rceil) \cdot e^{-\gamma/2}. \tag{5.105}$$

Further, by Lemmas 5.5.20 and 5.5.27,

$$\mathbb{P}[\mathcal{E} | \mathcal{E}_1, N^t = C, N^{t-1} = C'] \geq \frac{1}{128} - 31n \cdot e^{-\gamma/2}.$$

This gives the lemma when combined with (5.105). □

5.5.6 Completing the Analysis

By combining the nine cases of Lemmas 5.5.20, 5.5.21, 5.5.22, 5.5.23, 5.5.24, 5.5.25, 5.5.26, 5.5.27, and 5.5.28 we can conclude the following general statement:

Lemma 5.5.29 ($O(1)$ Step Convergence From Typical Configurations)**.** *Assume the input execution α_X of $\mathcal{L}_{n,\gamma}$ has X^t fixed for all t and that $\|X^t\|_1 \geq 1$. Consider any pair of typical configurations (Definition 5.5.9) C', C with $C(X) = C'(X) = X^t$. For any time $t \geq 1$,*

$$\mathbb{P}[(N^{t+i}, N^{t+i+1}) \textit{ is near-stable for some } i \leq 11 | N^t = C, N^{t-1} = C'] \geq \frac{1}{128} - 33ne^{-\gamma/2}.$$

Proof. The nine cases of the above listed lemmas cover all possible pairs of typical configurations C, C'. So the lemma follows immediately. □

From Lemma 5.5.29 we can conclude an even more general result:

Lemma 5.5.30 (General $O(1)$ Step Convergence)**.** *Assume the input execution α_X of $\mathcal{L}_{n,\gamma}$ has X^t fixed for all t and that $\|X^t\|_1 \geq 1$. Consider any pair of configurations C', C with $C(X) = C'(X) = X^t$. For any time $t \geq 1$,*

$$\mathbb{P}[(N^{t+i}, N^{t+i+1}) \text{ is near-stable for some } i \leq 13 | N^t = C, N^{t-1} = C'] \geq \frac{1}{128} - 39ne^{-\gamma/2}.$$

Proof. By Corollary 5.5.10, for any C', C,

$$\begin{aligned}\mathbb{P}[N^{t+1}, N^{t+2} \text{ are typical } | N^t = C, N^{t-1} = C'] &\geq 1 - 2(n + \lceil \log_2 n \rceil + 1) \cdot e^{-\gamma/2}\\ &\geq 1 - 6n \cdot e^{-\gamma/2}.\end{aligned}$$

The lemma then follows by combining this bound with Lemma 5.5.29. □

Finally, using Lemma 5.5.30, we can prove the two main theorems of this section. The proofs are similar to those of the main theorems for the two-inhibitor network in Section 5.3.7.

Theorem 5.5.2 ($O(\log n)$-Inhibitor WTA)**.** *For $\gamma \geq 12 \ln(39 t_s n/\delta)$, $\mathcal{L}_{n,\gamma}$ solves* WTA(n, t_c, t_s, δ) *for any $t_c \geq 2086(\log_2(1/\delta) + 1)$. $\mathcal{L}_{n,\gamma}$ contains $\lceil \log_2 n \rceil + 1$ auxiliary inhibitors.*

Proof. Consider $\mathcal{L}_{n,\gamma}$ starting with any initial configurations N^0, N^1 and given an infinite input execution α_X with X^t fixed for all t. We consider two cases:

Case 1: $\|X^t\|_1 \geq 1$.

Let $\Delta = 15$ and $r = 139(\log_2(1/\delta) + 1)$. Let $\mathcal{E}$ be the event that there is some time $t \leq t_c$ where (N^t, N^{t+1}) is a near-stable pair of configurations.

For any $i \geq 0$, let $\mathcal{E}_i$ be the event that there is some time $t \in \{i\Delta + 1, ..., (i+1)\Delta - 1\}$ where (N^t, N^{t+1}) is a near-stable pair of configurations. By Lemma 5.5.30 we have:

$$\mathbb{P}[\mathcal{E}_i | N^{i\Delta}] \geq \frac{1}{128} - 39n \cdot e^{-\gamma/2} \geq \frac{1}{200}$$

by the requirement that $\gamma > 12 \ln(39 t_s n/\delta)$. Let $Z_0, ..., Z_{r-1} \in \{0, 1\}$ be independent coin flips, with $\mathbb{P}[Z_i = 1] = 1/200$. Applying Lemma 5.2.3:

$$\mathbb{P}[\mathcal{E}] = \mathbb{P}\left[\bigcap_{i=0}^{r-1} \mathcal{E}_i\right] \geq \mathbb{P}\left[\sum_{i=0}^{r-1} Z_i \geq 1\right] = 1 - \left(\frac{199}{200}\right)^r.$$

Using that $r = 139(\log_2(1/\delta)+1)$ and that $\left(\frac{199}{200}\right)^{139} \leq \frac{1}{2}$:

$$\mathbb{P}[\mathcal{E}] \geq 1 - \left(\frac{199}{200}\right)^{139(\log_2(1/\delta)+1)} \geq 1 - \frac{\delta}{2}. \tag{5.106}$$

Thus, with probability $\geq 1 - \frac{\delta}{2}$ there is some time $t \leq r \cdot \Delta - 1 \leq 2085 \cdot (\log_2(1/\delta) + 1) - 1 \leq t_c - 2$ in which (N^t, N^{t+1}) is a near-stable pair of configurations. By Corollary 5.5.19, if (N^t, N^{t+1}) is a near-stable pair, with probability $\geq 1 - 3t_s n \cdot e^{-\gamma/2}$, N^{t+2} is a valid WTA configuration and:

$$Y^{t+2} = Y^{t+3} = ... = Y^{t+2+t_s}.$$

By the requirement that $\gamma \geq 12\ln(39t_s n/\delta)$ and (5.106) we thus have that the network reaches a valid WTA configuration within time t_c and remains in it for time t_s with probability at least:

$$\left(1 - \frac{\delta}{2}\right) \cdot \left(1 - \frac{\delta}{e^6}\right) \geq 1 - \delta,$$

yielding the theorem in this case.

Case 0: $\|X^t\|_1 = 0$

In this case, by Corollary 5.5.6 and a union bound, for any configurations C, C' with $C(X) = C'(X) = X^t$:

$$\begin{aligned}\mathbb{P}[\|Y^2\|_1 = ... = \|Y^{2+t_s}\|_1 = 0 | N^1 = C, N^0 = C'] &\geq 1 - 3t_s n \cdot e^{-\gamma/2}\\ &\geq 1 - \delta\end{aligned}$$

by the requirement that $\gamma \geq 12\ln(39t_s n/\delta)$. This easily gives the theorem in this case since when $\|X^t\|_1 = 0$, $\|Y^t\|_1 = 0$ is a valid WTA output configuration.

□

We now use a similar argument to show with what parameters $\mathcal{L}_{n,\gamma}$ solves the expected-time WTA problem of Definition 5.2.9.

Theorem 5.5.3 ($O(\log n)$-Inhibitor Expected-Time WTA)**.** *For $\gamma \geq 12\ln(39t_s n)$, $\mathcal{L}_{n,\gamma}$ solves* WTA-EXP(n, t_c, t_s) *for any $t_c \geq 4001$. $\mathcal{L}_{n,\gamma}$ contains $\lceil \log_2 n \rceil + 1$ auxiliary inhibitors.*

Proof. Our proof closely follows that of Theorem 5.3.3. Recall that in Definition 5.2.9 we defined the convergence time for any infinite input execution α_X and output execution α_Y:

$t(\alpha_X, t_s, \alpha_Y) = \min\left\{t : Y^t \text{ is a valid WTA output configuration for } X^t \text{ and } Y^t = ... = Y^{t+t_s}\right\}.$

Define the worst case expected convergence time of $\mathcal{L}_{n,\gamma}$ on input α_X by:

$$t_{max}(\alpha_X) = \max_{\alpha^1 = N^0 N^1}\left(\mathop{\mathbb{E}}_{\alpha_Y \sim \mathcal{D}_Y(\mathcal{L}_{n,\gamma},\alpha^1,\alpha_X)} t(\alpha_X, t_s, \alpha_Y)\right).$$

To prove the lemma we must prove that for any α_X with X^t fixed for all t, $t_{max}(\alpha_X) \leq 16$. Fixing such an α_X, for any initial configuration $\alpha^1 = N^0 N^1$, let $\mathop{\mathbb{E}}_{\alpha^1}$ and $\mathop{\mathbb{P}}_{\alpha^1}$ denote the expectation and probability of an event taken over executions drawn from $\mathcal{D}(\mathcal{L}_{n,\gamma}, \alpha^1, \alpha_X)$.

We consider the case when α_X has $\|X^t\|_1 \geq 1$. The case when $\|X^t\|_1 = 0$ follows easily from a similar proof using Corollary 5.5.6.

Let $\Delta = 15$ and let $\mathcal{E}_1$ be the event that there is some $t \in \{1, ..., \Delta - 1\}$ where (N^t, N^{t+1}) is a near-stable pair of configurations. Let $\mathcal{E}_{stab}$ be the event that there is some $t \in \{1, ..., \Delta - 1\}$ where (N^t, N^{t+1}) is a near-stable pair of configurations and additionally, where N^{t+2} is a valid WTA output configuration with $N^{t+2} = ... = N^{t+2+t_s}$. Let $\bar{\mathcal{E}}_1$ and $\bar{\mathcal{E}}_{stab}$ be the complements of these two events. By Lemma 5.5.30, for any initial $\alpha^1 = N^0 N^1$:

$$\mathop{\mathbb{P}}_{\alpha^1}[\mathcal{E}_1] \geq \frac{1}{128} - 39ne^{-\gamma/2} \geq \frac{1}{200}. \tag{5.107}$$

by the requirement that $\gamma \geq 12\ln(39 t_s n)$. Further, by Corollary 5.5.19, if (C', C) is a near-stable pair of configurations, then

$$\begin{aligned}\mathop{\mathbb{P}}_{\alpha^1}[N^{t+1} \text{ is a valid WTA output config. and } N^{t+2} = ... = N^{t+2+t_s} | N^t = C', N^{t+1} = C]\\ \geq 1 - 3t_s ne^{-\gamma/2}\\ \geq 1 - \frac{1}{1000t_s}\end{aligned} \tag{5.108}$$

where the bound holds since $\gamma \geq 12\ln(39t_s n) \geq 6\ln(39t_s^2 n)$ and so $e^{-\gamma/2} \leq \frac{1}{(39nt_s^2)^6} \leq \frac{1}{1000nt_s^2}$. Together (5.107) and (5.108) give that:

$$\mathop{\mathbb{P}}_{\alpha^1}[\mathcal{E}_{stab}] \geq \mathop{\mathbb{P}}_{\alpha^1}[\mathcal{E}_{stab}|\mathcal{E}_1] \cdot \mathop{\mathbb{P}}_{\alpha^1}[\mathcal{E}_1] \geq \frac{1}{200}\cdot\left(1 - \frac{1}{1000t_s}\right) \geq \frac{1}{200} - \frac{1}{1000t_s}.$$

We can write:

$$\begin{aligned}\mathop{\mathbb{E}}_{\alpha^1}[t(\alpha_X, t_s, \alpha_Y)] = &\mathop{\mathbb{E}}_{\alpha^1}[t(\alpha_X, t_s, \alpha_Y)|\mathcal{E}_{stab}] \cdot \mathop{\mathbb{P}}_{\alpha^1}[\mathcal{E}_{stab}] \\ &+ \mathop{\mathbb{E}}_{\alpha^1}[t(\alpha_X, t_s, \alpha_Y)|\mathcal{E}_1, \bar{\mathcal{E}}_{stab}] \cdot \mathop{\mathbb{P}}_{\alpha^1}[\mathcal{E}_1, \bar{\mathcal{E}}_{stab}] \\ &+ \mathop{\mathbb{E}}_{\alpha^1}[t(\alpha_X, t_s, \alpha_Y)|\bar{\mathcal{E}}_1] \cdot \mathop{\mathbb{P}}_{\alpha^1}[\bar{\mathcal{E}}_1]\end{aligned} \tag{5.109}$$

Conditioned on $\mathcal{E}_{stab}$ (which also requires that $\mathcal{E}_1$ occurs), the network reaches a near-stable pair of configuration within Δ steps, reaches a valid WTA output configuration within $\Delta + 1$ steps, and stabilizes for t_s steps. Thus, we have:

$$\mathop{\mathbb{E}}_{\alpha^1}[t(\alpha_X, t_s, \alpha_Y)|\mathcal{E}_{stab}] \leq \Delta + 1.$$

Conditioned on $\mathcal{E}_1, \bar{\mathcal{E}}_{stab}$ the network reaches a near-stable pair of configurations, but does not stabilize. We can bound

$$\begin{aligned}\mathop{\mathbb{E}}_{\alpha^1}[t(\alpha_X, t_s, \alpha_Y)|\mathcal{E}_1, \bar{\mathcal{E}}_{stab}] &\leq (\Delta + 1 + t_s) + \mathop{\mathbb{E}}_{N^{\Delta+t_s} N^{\Delta+1+t_s}}[t(\alpha_X, t_s, \alpha_Y)] \\ &\leq \Delta + 1 + t_s + t_{max}(\alpha_X).\end{aligned}$$

Finally, conditioned on $\bar{\mathcal{E}}_1$, the network does not reach a pair of near-stable configurations within Δ steps. We have:

$$\begin{aligned}\mathop{\mathbb{E}}_{\alpha^1}[t(\alpha_X, t_s, \alpha_Y)|\bar{\mathcal{E}}_1] &\leq \Delta + \mathop{\mathbb{E}}_{N^{\Delta-1} N^{\Delta}}[t(\alpha_X, t_s, \alpha_Y)] \\ &\leq \Delta + t_{max}(\alpha_X).\end{aligned}$$

We can plug these bounds along with the probability bounds of (5.107) and (5.108) into (5.109) to obtain:

$$\begin{aligned}\mathop{\mathbb{E}}_{\alpha^1}[t(\alpha_X, t_s, \alpha_Y)] &\leq (\Delta + 1) \cdot \left(\frac{1}{200} - \frac{1}{1000t_s}\right) + (\Delta + 1 + t_{max}(\alpha_X) + t_s) \cdot \frac{1}{1000t_s} \\ &\quad + (\Delta + t_{max}(\alpha_X)) \cdot \frac{199}{200} \\ &\leq \Delta + 1 + t_{max}\left(\frac{199}{200} + \frac{1}{1000t_s}\right) + \frac{t_s}{1000t_s} \\ &\leq \Delta + 1 + t_{max}(\alpha_X) \cdot \frac{249}{250} + \frac{1}{1000}.\end{aligned}$$

Since this bound holds for all α^1 we have:

$$t_{max}(\alpha_X) \leq \Delta + 1 + t_{max}(\alpha_X) \cdot \frac{249}{250} + \frac{1}{100}$$

which gives $t_{max}(\alpha_X) \leq 250(\Delta + 1) + \frac{250}{1000} \leq 250\Delta + 251$. This bound holds for all α_X and so gives the lemma, after recalling that $\Delta = 15$. □

5.5.7 Constructions With Runtime Tradeoffs

The family of two-inhibitor networks of Section 5.3 and the family of $O(\log n)$-inhibitor networks of Section 5.5.2 represent two extremes of a tradeoff between number of inhibitors and runtime. As shown in Theorem 5.4.3, the two-inhibitor network uses the minimum number of neurons required to solve WTA with a reasonably long stability period. At least when considering a constant probability of success, the $O(\log n)$-inhibitor networks gives optimal convergence time of $O(1)$ steps. In this section we outline, at a high level, two families of networks that allow a tradeoff between these two extremes.

Fixed Convergence Time Construction

The first construction lets us to achieve any desired convergence time θ if sufficiently many inhibitors are used. Specifically, we describe a family of networks which converge to a valid WTA output configuration with constant probability in $O(\theta)$ steps (and with probability $\geq 1 - \delta$ in $O(\theta \cdot \log 1/\delta)$ steps), using $O(\theta \log^{1/\theta} n)$ inhibitors. Note that setting $\theta = 1$ recovers the runtime-inhibitor tradeoff of our $O(\log n)$-inhibitor networks.

We have one stability inhibitor a_s that functions in the same way as the stability inhibitor in our $O(\log n)$-inhibitor construction (Definition 5.5.1), ensuring that, once the network reaches a valid WTA output configuration, it remains in this configuration for t_s consecutive time steps with high probability (as long as some other stability conditions, similar to the near-stable condition of Definition 5.5.16 are satisfied).

The remaining inhibitors are split into θ groups each containing $O(\log^{1/\theta} n)$ inhibitors. The i^{th} group for $i \in \{2, ..., \theta\}$ is responsible for reducing the number of competing outputs from any number

$$k \in \left(2^{\log^{(i-1)/\theta} n}, 2^{\log^{i/\theta} n}\right]$$

to some $k' \leq 2^{\log^{(i-1)/\theta} n}$. With $O(\log^{1/\theta} n))$ inhibitors this can be done with high probability in $O(1)$ time steps via a method described below. Thus, in $O(\theta)$ time steps, the number of firing outputs (corresponding to firing inputs will reduce from at most $2^{\log^{\theta/\theta} n} = n$ down to $2^{\log^{1/\theta} n}$. Once the number of computing outputs is this low, the final group of $O(\log^{1/\theta} n)$ inhibitors can drive convergence to WTA with constant probability in $O(1)$ steps using an identical strategy to that employed in our $O(\log n)$-inhibitor network. n is just replaced by $2^{\log^{1/\theta} n}$.

It remains to explain how the number of competing outputs is reduced from $k \in [2^{\log^{(i-1)/\theta} n}, 2^{\log^{i/\theta} n}]$ to $k' \leq 2^{\log^{(i-1)/\theta} n}$ in a single time step using $O(\log^{1/\theta} n)$ inhibitors. Again, we use a strategy similar to our $O(\log n)$-inhibitor construction. depending on the number of firing outputs at time t, the $O(\log^{1/\theta} n)$ inhibitors in group i induce $O(\log^{1/\theta} n)$ different firing probabilities. Letting $\Delta(j) = j \cdot \log^{(i-1)/\theta} n$, these firing probabilities are:

$$\frac{1}{2^{\Delta(1)}}, \frac{1}{2^{\Delta(2)}}, \ldots, \frac{1}{2^{\Delta((\log^{1/\theta} n - 1))}}, \frac{1}{2^{\Delta(\log^{1/\theta} n)}}.$$

Note that $\Delta(1) = \log^{(i-1)/\theta} n$ and $\Delta(\log^{1/\theta} n) = \log^{i/\theta} n$. Additionally, the ratio between adjacent probabilities is $2^{\log^{(i-1)/\theta} n}$. Thus, for any number $k \in \left(2^{\log^{(i-1)/\theta} n}, 2^{\log^{i/\theta} n}\right]$ of firing outputs at time t, the inhibitors can induce a firing probability between $\frac{1}{k}$ and $\frac{2^{\log^{(i-1)/\theta} n}}{k}$. Thus, with good probability, the number of firing outputs will reduce to some value $k' \leq 2^{\log^{(i-1)/\theta} n}$.

While we do not analyze this construction in detail, using similar proof techniques to those used for our $O(\log n)$-inhibitor construction, it is possible to show:

Theorem 5.5.31 (θ-Step WTA)**.** *For any $n \in \mathbb{Z}^{\geq 2}$ and $\theta \in \mathbb{Z}^{\geq 1}$ there is an a family of SNNs containing $O(\theta \cdot \log^{1/\theta} n)$ auxiliary inhibitory neurons which solve* $\text{WTA}(n, t_c, t_s, \delta)$ *for any t_s, δ and any $t_c \geq c_1 \cdot \theta(\log_2(1/\delta) + 1)$, where c_1 is some fixed constant.*

Fixed Inhibitor Budget Construction

Our second construction is a family of networks using α inhibitors for any $\alpha \geq 2$, which converges to a valid WTA state with constant probability in $O(\alpha \cdot (\log n)^{1/(\alpha-1)})$ steps (and with probability $\geq 1 - \delta$ in $O(\alpha \cdot (\log n)^{1/(\alpha-1)} \cdot \log 1/\delta)$ steps. Note that for $\alpha = 2$, this gives convergence in $O(\log n)$ steps, matching the performance of the two-inhibitor network construction of Section 5.3.

As in our two-inhibitor construction (Definition 5.3.1) and $O(\log n)$-inhibitor construction (Definition 5.5.1) we employ one stability inhibitor, which ensures that, once the network reaches a valid WTA output configuration, it remains in such a configuration with good probability for t_s consecutive steps (again, as long as some other stability conditions, similar to the near-stable condition of Definition 5.5.16 are satisfied).

We label the remaining $\alpha - 1$ inhibitors $a_1, ..., a_{\alpha-1}$. For each i, a_i fires with high probability at time $t+1$ whenever $k \geq 2^{(\log n)^{(i-1)/(\alpha-1)}}$ outputs fire at time t. In this way, with high probability, $a_1, ..., a_i$ fire (and all other inhibitors do not fire) at time $t+1$ whenever the number of firing outputs k is in the range:

$$R_i \overset{\text{def}}{=} \left[2^{(\log n)^{(i-1)/(\alpha-1)}}, 2^{(\log n)^{i/(\alpha-1)}}\right).$$

We set the inhibitory weights such that when a_i fires (along with a_j for all $j \leq i$), each firing output fires in the next step with probability $p_i = \frac{1}{2^{(\log n)^{(i-1)/(\alpha-1)}}}$. For $k \in R_i$ we have:

$$p_i \cdot k \in \left[1, 2^{(\log n)^{i/(\alpha-1)} \cdot \left(1-1/(\log n)^{1/(\alpha-1)}\right)}\right).$$

With each output continuing to fire with probability $\frac{1}{2^{(\log n)^{(i-1)/(\alpha-1)}}}$ in each step, starting from at most $2^{(\log n)^{i/(\alpha-1)}}$ competing outputs, we thus reach a configuration with $k \leq 2^{(\log n)^{(i-1)/(\alpha-1)}}$ firing outputs with good probability within $O((\log n)^{1/(\alpha-1)})$ steps. Overall, over α such levels, the network reaches a valid WTA configuration with constant probability in $O(\alpha \cdot (\log n)^{1/(\alpha-1)})$ steps.

Again, while we do not analyze this construction here, it is possible to prove the following:

Theorem 5.5.32 (α-Inhibitor WTA). *For any $n, \alpha \in \mathbb{Z}^{\geq 2}$ there is an a family of SNNs containing α auxiliary inhibitory neurons which solve* WTA(n, t_c, t_s, δ) *for any t_s, δ and any $t_c \geq c_1 \cdot \alpha(\log n)^{1/(\alpha-1)} \cdot (\log_2(1/\delta)+1)$, where c_1 is some fixed constant.*

5.6 Discussion and Future Work

We have presented an exploration of the WTA problem in stochastic spiking neural networks, giving network constructions, runtime analysis, and lower bounds. Our work leaves open a number of questions, both in regards to the WTA problem, and more broadly, in exploring spiking neural networks from an algorithmic perspective. We discuss some of these directions below. See also Section 5.2.5 in which we discuss possible extensions two and modifications of our basic spiking neural network model.

5.6.1 Winner-Take-All Extensions and Open Questions

We first overview future work directly related to the WTA problem studied in this chapter.

Lower Bounds

In Section 5.4 we present lower bounds for one and two-inhibitor WTA networks (Theorems 5.4.3 and 5.4.14) . In [LMP17a] we also give nearly tight lower bounds on the convergence time for networks using $\alpha > 2$ inhibitors, in a similar model to the one presented here. A few interesting open questions remain:

- Can our lower bound for two-inhibitor networks (Theorem 5.4.14) be tightened to match our upper bound (Theorem 5.3.2) up to a constant factor, rather than a $O(\log \log n)$ factor?

- Our lower bounds apply to somewhat restricted classes of simple and symmetric SNNs (Definitions 5.4.1 and 5.4.2 respectively.) We conjecture, however, that these lower bounds can be shown for general SNNs, with no restrictions on the network structure.

- It would be interesting to obtain lower bounds which help explain the relationship between history and convergence time. Our $O(\log n)$-inhibitor network of Section 5.5.2 achieves constant probability, $O(1)$ convergence time. However, it requires a history period of two steps. Is it possible to show that, with no history period (i.e., in the basic SNN model of Section 5.2), it is not possible to achieve this convergence time? Is the $O(\log n)$ time for our two-inhibitor network optimal for historyless networks, regardless of the number of inhibitors used? In a network with history, is it possible to solve WTA with a single auxiliary neuron, or can the lower bound of Theorem 5.4.3 be extended to this case?

Variations on the WTA Problem

Our work focuses on a simplified WTA problem in which all inputs either fire at every time step or never fire (see Definition 5.2.7). The challenge in solving this problem is in breaking symmetry between the firing inputs. In future work we plan to extend this work to consider more general variants of WTA, discussed below.

- **Non-Binary WTA:** We plan to extend our WTA work to a *non-binary* setting, in which inputs have different firing rates and selection is based on those rates. The most common requirement is for the network to select an input neuron with the highest or near-highest firing rate [YG89, CGL92, OL06].

  In the most basic setting, we can consider n inputs $x_1, ..., x_n$ with firing rates $r_1, ..., r_n \in [0, 1]$ as neurons that fire independently at random at each time step, each time with probability r_i. In [LMP17c], we presented an initial exploration of the non-binary WTA problem with inputs of this type, giving a solution using $O(n \log n)$ auxiliary neurons to select an input with firing rate within a constant factor of the maximum. It seems that in our basic SNN model, $\Omega(n \log n)$ auxiliary neurons may in fact be necessary: essentially, for each input neuron, $\Omega(\log n)$ auxiliary neurons are needed to record a short firing history, from which the firing probability r_i can be (implicitly) estimated to within a constant factor, with high probability. Thus, networks using a sublinear number of auxiliary neurons, like those we gave for binary WTA, may be ruled out.

  However, more efficient solutions may be possible if we use an SNN model with a history period. Even without history, efficient solutions may be possible if we relax or modify the WTA problem. For example, we may just require selecting each neuron with probability $\frac{r_i}{\sum_{i=1}^{n} r_i}$, or some other function of the firing probabilities.

- **WTA with Different Output Conditions:** It would be interesting to consider variations on the basic output condition of the WTA problem, both in the binary and non-binary input setting. For example, k-WTA is a common variant of WTA, in which the goal is to select k inputs with the highest or near-highest firing rates [MEAM89, Maa96, WS03, HW08], rather than just as single input. k-WTA, for example, has been used in recent work in modeling sparse coding in the fly olfactory via random projection methods [DSN17]. It would be interesting to understand the fundamental complexity (in terms of network complexity and convergence time) of implementing this primitive in spiking neural networks.

Applications of WTA

Finally, it would be interesting to explore how our WTA constructions can be integrated into solutions to higher-level neural tasks. Some problems which may be worth considering are:

- Attention in sensory processing systems, which is thought to be implemented via WTA competition [LIKB99, IK01]. In this setting, a specific set of neurons is activated while others are suppressed, allowing computation to happen on the selected neurons without interference. Formulating and studying simple tasks which model the use of WTA as an attention mechanism would be an interesting direction.

- The use of WTA in neural sparse coding algorithms [DSN17]. In this setting, a stimulus, such as a specific odor, triggers the firing of a large set of neurons, which is then "sparsified" via non-binary k-WTA competition to k neurons that encode the odor – those with the strongest response to the stimulus.

- Clustering problems, in a which the network is set up such that, when given any input, the neuron which corresponds to the best cluster assignment for this input fires at the highest rate. Non-binary WTA can be used to select this neuron [AF94, WR94].

5.6.2 Other Neural Computational Primitives

WTA is a basic primitive that seems to be useful in a broad range of neural tasks, discussed above. It is also useful in exploring the computational power of SNNs and studying tradeoffs among such costs as runtime, network size, and the use of randomness. In future work, we hope to to identify other such basic primitives, as a means of building a general algorithmic theory for spiking neural networks. We discuss some possible directions below:

Neural Indexing

In [LMP17b] we define and study the *Neuro-RAM* problem: the network is given a set of n binary inputs X_1, along with a smaller set of $\log n$ inputs X_2 whose firing pattern represents an index into X_1. The goal is the for the output to fire if and only if the selected neuron in X_1 is firing. In [LMP17b], we applied our Neuro-RAM primitive to the similarity testing problem: given two binary inputs X_1, X_2, determine with high probability whether their Hamming distance is greater than some threshold. We solved this problem efficiently by using a Neuro-RAM model to select and compare random positions in X_1 and X_2.

It would be interesting to formulate other basic neural tasks that may employ a Neuro-RAM primitive. For example, one can consider estimating the differences between firing patterns (e.g., representing sensory inputs at successive times), estimating the average firing activity of a group of neurons responding to some stimulus, or randomly “exploring” different sets of neurons whose firing may trigger a desired response (such as the recall of a memory).

In our work, aside from efficient Neuro-RAM constructions, we give lower bounds demonstrating that these constructions give a near-optimal tradeoff between network size and runtime in our basic SNN model with a sigmoidal spike probability function. Our proofs are based on reducing an SNN to a distribution over deterministic networks, and then bounding the VC dimension [Vap98] of these networks.

Generally, while VC dimension has been studied for spiking networks [ZP96, MS99], we are the first to apply it to prove computational lower bounds. We hope to continue our work, greatly expanding the toolkit for proving lower bounds in biologically plausible networks, including those with different spike probability functions, with history, and even with refractory periods or spike propagation delays.

Further, Neuro-RAM lower bounds let us compare our SNN models with artificial neural network models that use continuous-output gates, e.g., sigmoidal gates [Bar93, Bar97] or rectified linear units (ReLUs) [JKLR09, NH10]. The Neuro-RAM problem can be solved very efficiently in such networks [Koi96], contrasting with its apparent difficulty in SNNs. However, the continuous-output solutions do not seem very robust: they do not appear to tolerate small variations in edge weights, and they seem to rely on precise (infinite precision) gate outputs. We hope that comparing these models with our SNNs could lead to a better understanding of continuous-output circuits in noisy settings. We conjecture that in these settings, these networks behave more like their spiking network counterparts than like ideal continuous-output networks.

Binary Vector Problems

Beyond the WTA and Neuro-RAM problems it would also be interesting to consider other basic problems with fixed input firing patterns, that is, binary vector problems. For example, the problem of estimating the total firing strength of a population of neurons corresponds to determining the Hamming weight of a binary input vector. More complex problems include string matching, in which we test whether the input firing pattern contains a copy of a specified substring. In the brain, this may be used for pattern recognition, or for alignment and comparison of patterns triggered by multiple stimuli.

Problems with Non-Binary Inputs

It would also be valuable to consider problems in which the input is non-binary, for example, problems that require computing some function of the firing rates of the input neurons. For example, we may consider the problem of approximating the sum of input firing probabilities, or more generally, some norm of the firing rate vector $r_1, ..., r_n$. We may also consider similarity testing under different norms (generalizing the Hamming distance similarity testing problem in [LMP17b]).

Many algorithms proposed in computational neuroscience and computer science apply operations such as addition and multiplication to continuous neuron output values [DR89, AZGMS14, AGMM15, DSN17]. These continuous output values can be thought of as abstractions of firing rates in SNNs. We would like to understand how such basic operations may be emulated in more biologically plausible SNNs.

Synchronization Problems

Finally, it would be interesting to study algorithmic primitives that are not expressed in terms of simple input-output mappings. Notably, we would like to consider synchronization problems in which neurons, starting from arbitrary firing states, cooperate to align their firing in some way. Synchronization of spiking patterns and the emergence of neural rhythms are widely studied in both empirical and computational neuroscience [VVAE94, Buz06, RGDA97, WSO$^+$07]. Synchronization is also an important technique used in distributed algorithms [BL85, KO87, Cri89, LSW09], and thus is a natural problem to include in an algorithmic theory of neural computation.

5.6.3 Learning Problems and Dynamic Networks

The work presented in this chapter, as well as the work proposed in Section 5.6.2 focuses on computation in *static* networks, with fixed edge weights. An important direction is studying algorithms for learning in *dynamic spiking networks*, in which synapse weights change throughout the computation via, for example, a Hebbian update rule.

In classical Hebbian learning [Heb05, CD08], the weight of a synapse is continuously updated by a factor that depends on the product of the firing strengths of its two endpoints. The more the firing of the endpoints correlates, the stronger the synapse becomes. We hope to define a Hebbian-style rule for our synchronous SNN models similar to, for example, the simple rule used in [LMPV18]. When both endpoints fire at the same time, the synapse weight should increase by a small factor, and when just one fires, the weight may decrease. We then hope to use our dynamic network model to study the costs of many learning problems such as memory formation and concept association [Ama77, Val05, LMPV18], linear classification [HNGS+06], principal component analysis [San89, HO97, HO00], and sparse coding [OF04, AGMM15].

5.6.4 Neural Linear Algebraic Computation

Finally, a very interesting open direction is to tie together the two parts of this thesis, understanding how linear algebraic computation may be performed in spiking neural networks, possibly with the use of randomized computation. As discussed in Chapter 2, randomized algorithms have recently led to a number of breakthroughs in fundamental linear-algebraic and geometric problems such as linear regression [CW13, CLM+15], low-rank approximation [Sar06, NN13], clustering [BZD10], and locality-sensitive hashing [DIIM04, AI06].

Many mechanisms used in neural computation have close analogs to randomized linear algebraic techniques; e.g., Oja's Hebbian-style rule for neural principal component analysis (PCA) [HO97, HO00] is a common technique for low-memory PCA and eigenvector approximation [Sha15, JJK+16]. Fast linear-algebraic methods based on *random projections* [Ach03, Sar06] are also conjectured to play a role in neural computation [GS12a, AZGMS14], with random synaptic connectivity providing a natural implementation of randomized dimensionality-reduction. We hope to study the application of these techniques in stochastic spiking neural networks, forging connections with work on new algorithms for fast linear algebraic computation.

Bibliography

[AAB$^+$11] Yehuda Afek, Noga Alon, Omer Barad, Eran Hornstein, Naama Barkai, and Ziv Bar-Joseph. A biological solution to a fundamental distributed computing problem. *Science*, 331(6014):183–185, 2011.

[AB04] Chen Avin and Carlos Brito. Efficient and robust query processing in dynamic environments using random walk techniques. In *Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks*, pages 277–286. ACM, 2004.

[ABB00] Orly Alter, Patrick O Brown, and David Botstein. Singular value decomposition for genome-wide expression data processing and modeling. *Proceedings of the National Academy of Sciences*, 97(18):10101–10106, 2000.

[ABH16] Naman Agarwal, Brian Bullins, and Elad Hazan. Second order stochastic optimization in linear time. In *Proceedings of the 33rd International Conference on Machine Learning (ICML)*, 2016.

[Ach03] Dimitris Achlioptas. Database-friendly random projections: Johnson-Lindenstrauss with binary coins. *Journal of Computer and System Sciences*, 66(4):671–687, 2003.

[ACW16] Haim Avron, Kenneth L. Clarkson, and David P. Woodruff. Sharper bounds for regression and low-rank approximation with regularization, 2016.

[Ada90] Eldridge S Adams. Boundary disputes in the territorial ant *Azteca trigona*: effects of asymmetries in colony size. *Animal Behaviour*, 39(2):321–328, 1990.

[AF94] Hazem M Abbas and Moustafa M Fahmy. Neural networks for maximum likelihood clustering. *Signal Processing*, 36(1):111–126, 1994.

[AF02] David Aldous and Jim Fill. Reversible Markov chains and random walks on graphs, 2002.

[AFK$^+$01] Yossi Azar, Amos Fiat, Anna R. Karlin, Frank McSherry, and Jared Saia. Spectral analysis of data. In *Proceedings of the 33rd Annual ACM Symposium on Theory of Computing (STOC)*, pages 619–626, 2001.

[AFKM01] Dimitris Achlioptas, Amos Fiat, Anna R. Karlin, and Frank McSherry. Web search via hub synthesis. In *Proceedings of the 42nd Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 500–509, 2001.

[AGMM15] Sanjeev Arora, Rong Ge, Tengyu Ma, and Ankur Moitra. Simple, efficient, and neural algorithms for sparse coding. In *Proceedings of the 28th Annual Conference on Computational Learning Theory (COLT)*, pages 113–149, 2015.

[AGR16] Nima Anari, Shayan Oveis Gharan, and Alireza Rezaei. Monte Carlo Markov chain algorithms for sampling strongly Rayleigh distributions and determinantal point processes. In *Proceedings of the 29th Annual Conference on Computational Learning Theory (COLT)*, pages 103–115, 2016.

[AHS85] David H Ackley, Geoffrey E Hinton, and Terrence J Sejnowski. A learning algorithm for Boltzmann machines. *Cognitive Science*, 9(1):147–169, 1985.

[AI06] Alexandr Andoni and Piotr Indyk. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. In *Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 459–468, 2006.

[AKM$^+$17] Haim Avron, Michael Kapralov, Cameron Musco, Christopher Musco, Ameya Velingker, and Amir Zandieh. Random Fourier features for kernel ridge regression: Approximation bounds and statistical guarantees. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, 2017.

[AM05] Dimitris Achlioptas and Frank McSherry. On spectral learning of mixtures of distributions. In *Proceedings of the 18th Annual Conference on Computational Learning Theory (COLT)*, pages 458–469, 2005.

[AM07] Dimitris Achlioptas and Frank McSherry. Fast computation of low-rank matrix approximations. *Journal of the ACM*, 54(2), 2007.

[AM15a] Ahmed Alaoui and Michael W Mahoney. Fast randomized kernel ridge regression with statistical guarantees. In *Advances in Neural Information Processing Systems 28 (NIPS)*, pages 775–783, 2015.

[AM15b] Ahmed El Alaoui and Michael W. Mahoney. Fast randomized kernel methods with statistical guarantees. In *Advances in Neural Information Processing Systems 28 (NIPS)*, 2015.

[Ama77] S-I Amari. Neural theory of association and concept-formation. *Biological Cybernetics*, 26(3):175–185, 1977.

[AMS01] Dimitris Achlioptas, Frank Mcsherry, and Bernhard Schölkopf. Sampling techniques for kernel methods. In *Advances in Neural Information Processing Systems 14 (NIPS)*, 2001.

[And03] Alex M Andrew. Spiking neuron models: Single neurons, populations, plasticity. *Kybernetes*, 32(7/8), 2003.

[ANW14] Haim Avron, Huy Nguyen, and David Woodruff. Subspace embeddings for the polynomial kernel. In *Advances in Neural Information Processing Systems 27 (NIPS)*, pages 2258–2266, 2014.

[AS94] Christina Allen and Charles F Stevens. An evaluation of causes for unreliability of synaptic transmission. *Proceedings of the National Academy of Sciences*, 91(22):10380–10383, 1994.

[ASNN$^+$15] Maruan Al-Shedivat, Rawan Naous, Emre Neftci, Gert Cauwenberghs, and Khaled N Salama. Inherently stochastic spiking neurons for probabilistic neural computation. In *2015 7th International IEEE/EMBS Conference on Neural Engineering (NER)*, pages 356–359. IEEE, 2015.

[AV07] David Arthur and Sergei Vassilvitskii. k-means++: The advantages of careful seeding. In *Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1027–1035, 2007.

[AZGMS14] Zeyuan Allen-Zhu, Rati Gelashvili, Silvio Micali, and Nir Shavit. Sparse sign-consistent Johnson–Lindenstrauss matrices: Compression with neuroscience-based constraints. *Proceedings of the National Academy of the Sciences*, 2014.

[Bar93] Andrew R Barron. Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Transactions on Information Theory*, 39(3):930–945, 1993.

[Bar97] Peter L Bartlett. For valid generalization the size of the weights is more important than the size of the network. In *Advances in Neural Information Processing Systems 10 (NIPS)*, pages 134–140, 1997.

[BCF09] Constantine Bekas, Alessandro Curioni, and I Fedulova. Low cost high performance uncertainty quantification. In *Proceedings of the 2nd Workshop on High Performance Computational Finance*, page 8. ACM, 2009.

[BCKY16] Vladimir Braverman, Stephen R Chestnut, Robert Krauthgamer, and Lin F Yang. Sketches for matrix norms: Faster, smaller and more general. *arXiv:1609.05885*, 2016.

[BDN15] Jean Bourgain, Sjoerd Dirksen, and Jelani Nelson. Toward a unified theory of sparse dimensionality reduction in Euclidean space. In *Proceedings of the 47th Annual ACM Symposium on Theory of Computing (STOC)*, pages 499–508, 2015.

[BFKN18] Lucas Boczkowski, Ofer Feinerman, Amos Korman, and Emanuele Natale. Limits for rumor spreading in stochastic populations. In *Proceedings of the 9th Conference on Innovations in Theoretical Computer Science (ITCS)*, 2018.

[BHOP18] Anna Ben-Hamou, Roberto I Oliveira, and Yuval Peres. Estimating graph parameters via random walks with restarts. In *Proceedings of the 29th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1702–1714, 2018.

[BIS17] Arturs Backurs, Piotr Indyk, and Ludwig Schmidt. On the fine-grained complexity of empirical risk minimization: Kernel methods and neural networks. In *Advances in Neural Information Processing Systems 30 (NIPS)*, pages 4311–4321, 2017.

[BJ02] Francis Bach and Michael I. Jordan. Kernel independent component analysis. *Journal of Machine Learning Research*, 3(Jul):1–48, 2002.

[BK96] András A Benczúr and David R Karger. Approximating st minimum cuts in (n^2) time. In *Proceedings of the 28th Annual ACM Symposium on Theory of Computing (STOC)*, pages 47–55, 1996.

[BL85] James E Burns and Nancy A Lynch. The Byzantine firing squad problem. Technical report, Massachusetts Institute of Technology, 1985.

[BMF$^+$00] Wolfram Burgard, Mark Moors, Dieter Fox, Reid Simmons, and Sebastian Thrun. Collaborative multi-robot exploration. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA).*, volume 1, pages 476–481, 2000.

[BMV12] Vincenzo Bonifaci, Kurt Mehlhorn, and Girish Varma. Physarum can compute shortest paths. *Journal of Theoretical Biology*, 309:121–133, 2012.

[BP73] LA Bassalygo and MS Pinsker. The complexity of an optimal non-blocking commutation scheme without reorganization. *Problemy Peredaci Informacii*, 9(1):84–87, 1973.

[BP14] Monami Banerjee and Nikhil R Pal. Feature selection with SVD entropy: some modification and extension. *Information Sciences*, 264:118–134, 2014.

[BS83] Walter Baur and Volker Strassen. The complexity of partial derivatives. *Theoretical computer science*, 22(3):317–330, 1983.

[BS02] Maxim A Batalin and Gaurav S Sukhatme. Spreading out: A local approach to multi-robot coverage. In *Distributed Autonomous Robotic Systems*, pages 373–382. Springer, 2002.

[Buz06] Gyorgy Buzsaki. *Rhythms of the Brain*. Oxford University Press, 2006.

[BW09] Mohamed-Ali Belabbas and Patrick J. Wolfe. Spectral methods in machine learning: New strategies for very large datasets. *Proceedings of the National Academy of Sciences*, 106:369–374, 2009.

[BY02] Ziv Bar-Yossef. *The Complexity of Massive Data Set Computations*. PhD thesis, University of California at Berkeley, 2002.

[BZD10] Christos Boutsidis, Anastasios Zouzias, and Petros Drineas. Random projections for k-means clustering. In *Advances in Neural Information Processing Systems 23 (NIPS)*, 2010.

[Car14] Petre Caraiani. The predictive power of singular value decomposition entropy for stock market dynamics. *Physica A: Statistical Mechanics and its Applications*, 393:571–578, 2014.

[CC00] Trevor F Cox and Michael AA Cox. *Multidimensional Scaling*. CRC Press, 2000.

[CD08] Natalia Caporale and Yang Dan. Spike timing–dependent plasticity: a Hebbian learning rule. *Annual Review of Neuroscience*, 31:25–46, 2008.

[CDLN14] Alejandro Cornejo, Anna Dornhaus, Nancy Lynch, and Radhika Nagpal. Task allocation in ant colonies. In *International Symposium on Distributed Computing*, pages 46–60. Springer, 2014.

[CEM$^+$15] Michael B. Cohen, Sam Elder, Cameron Musco, Christopher Musco, and Madalina Persu. Dimensionality reduction for k-means clustering and low rank approximation. In *Proceedings of the 47th Annual ACM Symposium on Theory of Computing (STOC)*, 2015.

[CGK$^+$17] Flavio Chierichetti, Sreenivas Gollapudi, Ravi Kumar, Silvio Lattanzi, Rina Panigrahy, and David P Woodruff. Algorithms for ℓ_p low rank approximation. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, 2017.

[CGL92] Robert Coultrip, Richard Granger, and Gary Lynch. A cortical model of winner-take-all competition via lateral inhibition. *Neural Networks*, 5(1):47–54, 1992.

[Cha09] Bernard Chazelle. Natural algorithms. In *Proceedings of the 20th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 422–431, 2009.

[CKS11] Andrew Cotter, Joseph Keshet, and Nathan Srebro. Explicit approximations of the Gaussian kernel. *arXiv:1109.4603*, 2011.

[CLLM12] Kai-Min Chung, Henry Lam, Zhenming Liu, and Michael Mitzenmacher. Chernoff-Hoeffding bounds for Markov chains: Generalized and simplified. *arXiv:1201.0559*, 2012.

[CLM$^+$15] Michael B. Cohen, Yin Tat Lee, Cameron Musco, Christopher Musco, Richard Peng, and Aaron Sidford. Uniform sampling for matrix approximation. In *Proceedings of the 6th Conference on Innovations in Theoretical Computer Science (ITCS)*, 2015.

[CMKB04] Jorge Cortes, Sonia Martinez, Timur Karatas, and Francesco Bullo. Coverage control for mobile sensing networks. *IEEE Transactions on Robotics and Automation*, 20(2):243–255, 2004.

[CMM17] Michael B. Cohen, Cameron Musco, and Christopher Musco. Input sparsity time low-rank approximation via ridge leverage score sampling. In *Proceedings of the 28th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2017.

[CN85] Norishige Chiba and Takao Nishizeki. Arboricity and subgraph listing algorithms. *SIAM Journal on Computing*, 14(1):210–223, 1985.

[Coh16] Michael B. Cohen. Nearly tight oblivious subspace embeddings by trace inequalities. In *Proceedings of the 27th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 278–287, 2016.

[CP15] Michael B. Cohen and Richard Peng. l_p row sampling by Lewis weights. In *Proceedings of the 47th Annual ACM Symposium on Theory of Computing (STOC)*, 2015.

[CR09] Emmanuel J Candès and Benjamin Recht. Exact matrix completion via convex optimization. *Foundations of Computational Mathematics*, 9(6):717–772, 2009.

[CR12] Emmanuel J. Candès and Benjamin Recht. Exact matrix completion via convex optimization. *Communications of the ACM*, 55(6):111–119, 2012.

[Cri89] Flaviu Cristian. Probabilistic clock synchronization. *Distributed Computing*, 3(3):146–158, 1989.

[CW13] Kenneth L. Clarkson and David P. Woodruff. Low rank approximation and regression in input sparsity time. In *Proceedings of the 45th Annual ACM Symposium on Theory of Computing (STOC)*, 2013.

[CW17a] Kenneth L. Clarkson and David P. Woodruff. Low-rank PSD approximation in input-sparsity time. In *Proceedings of the 28th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2017.

[CW17b] Kenneth L. Clarkson and David P. Woodruff. Low-rank PSD approximation in input-sparsity time. In *Proceedings of the 28th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2061–2072, 2017.

[Dav91] Mark L Davison. Multidimensional scaling. 1991.

[DB11] Marco Dorigo and Mauro Birattari. Ant colony optimization. In *Encyclopedia of machine learning*, pages 36–39. Springer, 2011.

[Dem13] James Demmel. An arithmetic complexity lower bound for computing rational functions, with applications to linear algebra. *submitted to SIMAX*, 2013.

[DFK$^+$04] Petros Drineas, Alan M. Frieze, Ravi Kannan, Santosh Vempala, and V. Vinay. Clustering large graphs via the singular value decomposition. *Machine Learning*, 56(1-3):9–33, 2004.

[DHJZ15] Haishun Du, Qingpu Hu, Manman Jiang, and Fan Zhang. Two-dimensional principal component analysis based on Schatten p-norm for image feature extraction. *Journal of Visual Communication and Image Representation*, 32:55–62, 2015.

[DIIM04] Mayur Datar, Nicole Immorlica, Piotr Indyk, and Vahab S Mirrokni. Locality-sensitive hashing scheme based on p-stable distributions. In *Proceedings of the Twentieth Annual Symposium on Computational Geometry*, pages 253–262. ACM, 2004.

[DKJ$^+$07] Jason V Davis, Brian Kulis, Prateek Jain, Suvrit Sra, and Inderjit S Dhillon. Information-theoretic metric learning. In *Proceedings of the 24th International Conference on Machine Learning (ICML)*, pages 209–216, 2007.

[DKM06] Petros Drineas, Ravi Kannan, and Michael W Mahoney. Fast Monte Carlo algorithms for matrices I: Approximating matrix multiplication. *SIAM Journal on Computing*, 36(1):132–157, 2006.

[DKR02] Petros Drineas, Iordanis Kerenidis, and Prabhakar Raghavan. Competitive recommendation systems. In *Proceedings of the 34th Annual ACM Symposium on Theory of Computing (STOC)*, pages 82–90, 2002.

[DLWZ14] Xuefeng Duan, Jiaofen Li, Qingwen Wang, and Xinjun Zhang. Low rank approximation of the symmetric positive semidefinite matrix. *Journal of Computational and Applied Mathematics*, 260:236–243, 2014.

[DM05] Petros Drineas and Michael W Mahoney. On the Nyström method for approximating a Gram matrix for improved kernel-based learning. *Journal of Machine Learning Research*, 6:2153–2175, 2005.

[DMM06a] Petros Drineas, Michael W. Mahoney, and S. Muthukrishnan. Sampling algorithms for ℓ_2 regression and applications. In *Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1127–1136, 2006.

[DMM06b] Petros Drineas, Michael W Mahoney, and S Muthukrishnan. Subspace sampling and relative-error matrix approximation: Column-row-based methods. In *European Symposium on Algorithms*, pages 304–314. Springer, 2006.

[DMM08] Petros Drineas, Michael W Mahoney, and S Muthukrishnan. Relative-error CUR matrix decompositions. *SIAM Journal on Matrix Analysis and Applications*, 30(2), 2008.

[DR89] Richard Durbin and David E Rumelhart. Product units: A computationally powerful and biologically plausible extension to backpropagation networks. *Neural Computation*, 1(1):133–142, 1989.

[DRVW06] Amit Deshpande, Luis Rademacher, Santosh Vempala, and Grant Wang. Matrix approximation and projective clustering via volume sampling. In *Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2006.

[DSN17] Sanjoy Dasgupta, Charles F Stevens, and Saket Navlakha. A neural algorithm for a fundamental computing problem. *Science*, 358(6364):793–796, 2017.

[DTV11] Amit Deshpande, Madhur Tulsiani, and Nisheeth K. Vishnoi. Algorithms and hardness for subspace approximation. In *Proceedings of the 22nd Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 482–496, 2011.

[DV06] Amit Deshpande and Santosh Vempala. Adaptive sampling and fast low-rank matrix approximation. In *Approximation, Randomization, and Combinatorial Optimization: Algorithms and Techniques*, pages 292–303. Springer, 2006.

[ES02] Brian Everitt and Anders Skrondal. *The Cambridge Dictionary of Statistics*, volume 106. Cambridge University Press, 2002.

[ES09] Robert Elsässer and Thomas Sauerwald. Tight bounds for the cover time of multiple random walks. In *Automata, Languages and Programming*, pages 415–426. Springer, 2009.

[FHT08] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441, 2008.

[FK13] Ofer Feinerman and Amos Korman. Theoretical distributed computing meets biology: A review. In *International Conference on Distributed Computing and Internet Technology*, pages 1–18. Springer, 2013.

[FKLS12] Ofer Feinerman, Amos Korman, Zvi Lotker, and Jean-Sébastien Sereni. Collaborative search on the plane without communication. In *Proceedings of the 2012 ACM Symposium on Principles of Distributed Computing (PODC)*, pages 77–86, 2012.

[FKV04] Alan M. Frieze, Ravi Kannan, and Santosh Vempala. Fast Monte-Carlo algorithms for finding low-rank approximations. *Journal of the ACM*, 51(6):1025–1041, 2004.

[FMMS16] Roy Frostig, Cameron Musco, Christopher Musco, and Aaron Sidford. Principal component projection without principal component analysis. In *Proceedings of the 33rd International Conference on Machine Learning (ICML)*, 2016.

[FS02] Shai Fine and Katya Scheinberg. Efficient SVM training using low-rank kernel representations. *Journal of Machine Learning Research*, 2:243–264, 2002.

[FSS13] Dan Feldman, Melanie Schmidt, and Christian Sohler. Turning big data into tiny data: Constant-size coresets for k-means, PCA, and projective clustering. In *Proceedings of the 24th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1434–1453, 2013.

[FSW08] A Aldo Faisal, Luc PJ Selen, and Daniel M Wolpert. Noise in the nervous system. *Nature Reviews Neuroscience*, 9(4):292–303, 2008.

[FT07] Shmuel Friedland and Anatoli Torokhti. Generalized rank-constrained matrix approximations. *SIAM Journal on Matrix Analysis and Applications*, 29(2):656–659, 2007.

[GCD$^+$03] Sonja Glavaski, Madalena Chaves, Robert Day, Parthasarathi Nag, Anca Williams, and Wei Zhang. Vehicle networks: achieving regular formation. In *Proceedings of the American Control Conference*, volume 5, pages 4095–4100. IEEE, 2003.

[GE95] Ming Gu and Stanley C Eisenstat. A divide-and-conquer algorithm for the symmetric tridiagonal eigenproblem. *SIAM Journal on Matrix Analysis and Applications*, 16(1):172–191, 1995.

[GHJ$^+$16] Dan Garber, Elad Hazan, Chi Jin, Sham M Kakade, Cameron Musco, Praneeth Netrapalli, and Aaron Sidford. Faster eigenvector computation via shift-and-invert preconditioning. In *Proceedings of the 33rd International Conference on Machine Learning (ICML)*, 2016.

[Gil98] David Gillman. A Chernoff bound for random walks on expander graphs. *SIAM Journal on Computing*, 27(4):1203–1220, 1998.

[Git11] Alex Gittens. The spectral norm error of the naive Nyström extension. *arXiv:1110.5305*, 2011.

[GK02] Wulfram Gerstner and Werner M Kistler. *Spiking Neuron Models: Single Neurons, Populations, Plasticity.* Cambridge University Press, 2002.

[GKBM09] Minas Gjoka, Maciej Kurant, Carter T Butts, and Athina Markopoulou. A walk in Facebook: Uniform sampling of users in online social networks. *arXiv:0906.0060*, 2009.

[GL09] Ankur Gupta and Lyle N Long. Hebbian learning with winner take all for spiking neural networks. In *2009 International Joint Conference on Neural Networks*, pages 1054–1060. IEEE, 2009.

[GLF$^+$10] David Gross, Yi-Kai Liu, Steven T Flammia, Stephen Becker, and Jens Eisert. Quantum state tomography via compressed sensing. *Physical Review Letters*, 105(15):150401, 2010.

[GLMN09] Seth Gilbert, Nancy Lynch, Sayan Mitra, and Tina Nolte. Self-stabilizing robot formations over unreliable networks. *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, 4(3):17, 2009.

[GLPW16] Mina Ghashami, Edo Liberty, Jeff M. Phillips, and David P. Woodruff. Frequent directions: Simple and deterministic matrix sketching. *SIAM Journal on Computing*, 45(5):1762–1792, 2016.

[GM13] Alex Gittens and Michael Mahoney. Revisiting the Nyström method for improved large-scale machine learning. In *Proceedings of the 30th International Conference on Machine Learning (ICML)*, pages 567–575, 2013. Full version at arXiv:1303.1849.

[GMRL15] Mohsen Ghaffari, Cameron Musco, Tsvetomira Radeva, and Nancy Lynch. Distributed house-hunting in ant colonies. In *Proceedings of the 2015 ACM Symposium on Principles of Distributed Computing (PODC)*, 2015.

[Gor99] Deborah M Gordon. Interaction patterns and task allocation in ant colonies. In *Information Processing in Social Insects*, pages 51–67. Springer, 1999.

[GPT93] Deborah M Gordon, Richard E Paul, and Karen Thorpe. What is the function of encounter patterns in ant colonies? *Animal Behaviour*, 45(6):1083–1100, 1993.

[GS91] Leslie Greengard and John Strain. The fast Gauss transform. *SIAM Journal on Scientific and Statistical Computing*, 12(1):79–94, 1991.

[GS12a] Surya Ganguli and Haim Sompolinsky. Compressed sensing, sparsity, and dimensionality in neuronal information processing and data analysis. *Annual Review of Neuroscience*, 35:485–508, 2012.

[GS12b] Venkatesan Guruswami and Ali Kemal Sinop. Optimal column-based low-rank matrix reconstruction. In *Proceedings of the 23rd Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1207–1214. SIAM, 2012.

[Gu14] Ming Gu. Subspace iteration randomization and singular value problems. *arXiv:1408.2208*, 2014.

[GU17] François Le Gall and Florent Urrutia. Improved rectangular matrix multiplication using powers of the Coppersmith-Winograd tensor. *arXiv:1708.05622*, 2017.

[Gut92] Ivan Gutman. Total π-electron energy of benzenoid hydrocarbons. In *Advances in the Theory of Benzenoid Hydrocarbons II*, pages 29–63. Springer, 1992.

[Gut01] Ivan Gutman. The energy of a graph: old and new results. In *Algebraic Combinatorics and Applications*, pages 196–211. Springer, 2001.

[GVL12] Gene H. Golub and Charles F Van Loan. *Matrix Computations*, volume 3. JHU Press, 2012.

[GXL15] Rongbao Gu, Wei Xiong, and Xinjie Li. Does the singular value decomposition entropy have predictive power for stock market? Evidence from the Shenzhen stock market. *Physica A: Statistical Mechanics and its Applications*, 439:103–113, 2015.

[Har14] Moritz Hardt. Understanding alternating minimization for matrix completion. In *Proceedings of the 55th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 651–660, 2014.

[Hay09] Simon S Haykin. *Neural Networks and Learning Machines*, volume 3. Pearson, 2009.

[Heb05] Donald Olding Hebb. *The Organization of Behavior: A Neuropsychological Theory.* Psychology Press, 2005.

[Hig08] Nicholas J Higham. *Functions of Matrices: Theory and Computation.* SIAM, 2008.

[HJM13] Stefan Habenschuss, Zeno Jonke, and Wolfgang Maass. Stochastic computations in cortical microcircuit models. *PLoS Computational Biology*, 9(11):e1003311, 2013.

[HLM12] Moritz Hardt, Katrina Ligett, and Frank McSherry. A simple and practical algorithm for differentially private data release. In *Advances in Neural Information Processing Systems 25 (NIPS)*, pages 2348–2356. 2012.

[HNGS$^+$06] Ramón Huerta, Thomas Nowotny, Marta García-Sanchez, Henry DI Abarbanel, and Mikhail I Rabinovich. Learning classification in the olfactory system of insects. *Learning*, 16(8), 2006.

[HO97] Aapo Hyvärinen and Erkki Oja. One-unit learning rules for independent component analysis. In *Advances in Neural Information Processing Systems 10 (NIPS)*, pages 480–486, 1997.

[HO00] Aapo Hyvärinen and Erkki Oja. Independent component analysis: algorithms and applications. *Neural Networks*, 13(4):411–430, 2000.

[Hof03] Thomas Hofmann. Collaborative filtering via Gaussian probabilistic latent semantic analysis. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 259–266, 2003.

[HSW89] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, 1989.

[HT$^+$86] John J Hopfield, David W Tank, et al. Computing with neural circuits – a model. *Science*, 233(4764):625–633, 1986.

[HW08] Xiaolin Hu and Jun Wang. An improved dual neural network for solving a class of quadratic programming problems and its k-winners-take-all application. *IEEE Transactions on Neural Networks*, 19(12):2022–2031, 2008.

[IK01] Laurent Itti and Christof Koch. Computational modeling of visual attention. *Nature Reviews Neuroscience*, 2(3):194–203, 2001.

[IR77] Alon Itai and Michael Rodeh. Finding a minimum circuit in a graph. In *Proceedings of the 9th Annual ACM Symposium on Theory of Computing (STOC)*, pages 1–10, 1977.

[Isa08] Garth Isaak. Fast matrix multiplication and inversion, Math 242 notes. http://www.lehigh.edu/~gi02/m242/08linstras.pdf, 2008.

[Izh04] Eugene M Izhikevich. Which model to use for cortical spiking neurons? *IEEE Transactions on Neural Networks*, 15(5):1063–1070, 2004.

[JJK$^+$16] Prateek Jain, Chi Jin, Sham M Kakade, Praneeth Netrapalli, and Aaron Sidford. Streaming PCA: Matching matrix bernstein and near-optimal finite sample guarantees for Ojaś algorithm. In *Proceedings of the 29th Annual Conference on Computational Learning Theory (COLT)*, pages 1147–1164, 2016.

[JKLR09] Kevin Jarrett, Koray Kavukcuoglu, Yann LeCun, and Marc'Aurelio Ranzato. What is the best multi-stage architecture for object recognition? In *12th International Conference on Computer Vision*, pages 2146–2153. IEEE, 2009.

[JNS13] Prateek Jain, Praneeth Netrapalli, and Sujay Sanghavi. Low-rank matrix completion using alternating minimization. In *Proceedings of the 45th Annual ACM Symposium on Theory of Computing (STOC)*, pages 665–674, 2013.

[Jol02] Ian Jolliffe. *Principal Component Analysis.* Wiley Online Library, 2002.

[KBM12] Maciej Kurant, Carter T Butts, and Athina Markopoulou. Graph size estimation. *arXiv:1210.0460*, 2012.

[KK94] Samuel Kaski and Teuvo Kohonen. Winner-take-all networks for physiological models of competitive learning. *Neural Networks*, 7(6-7):973–984, 1994.

[KKKL11] Majid Khabbazian, Dariusz Kowalski, Fabian Kuhn, and Nancy Lynch. Decomposing broadcast algorithms using abstract MAC layers, 2011. http://groups.csail.mit.edu/tds/papers/Lynch/MIT-CSAIL-TR-2011-010.pdf.

[KKS65] V. V. Klyuev and N. I. Kokovkin-Shcherbak. Minimization of the number of arithmetic operations in the solution of linear algebra systems of equations. 5(1):25-43, 1965. translated by H. F. Cleaves.

[Kle99] Jon M. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5):604–632, 1999.

[KLM$^+$17] Michael Kapralov, Yin Tat Lee, Cameron Musco, Christopher Musco, and Aaron Sidford. Single pass spectral sparsification in dynamic streams. *SIAM Journal on Computing*, 46(1):456–477, 2017. Preliminary version in the IEEE Symposium on Foundations of Computer Science (FOCS), 2014.

[KLP$^+$16] Rasmus Kyng, Yin Tat Lee, Richard Peng, Sushant Sachdeva, and Daniel Spielman. Sparsified Cholesky and multigrid solvers for connection Laplacians. In *Proceedings of the 48th Annual ACM Symposium on Theory of Computing (STOC)*, 2016.

[KLS11] Liran Katzir, Edo Liberty, and Oren Somekh. Estimating sizes of social networks via biased sampling. In *Proceedings of the 20th International Conference on World Wide Web (WWW)*, pages 597–606. ACM, 2011.

[KLSC14] Liran Katzir, Edo Liberty, Oren Somekh, and Ioana A Cosma. Estimating sizes of social networks via biased sampling. *Internet Mathematics*, 10(3-4):335–359, 2014.

[KM09] Jonathan A. Kelner and Aleksander Madry. Faster generation of random spanning trees. In *Proceedings of the 50th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 13–21, 2009.

[KMT09] Sanjiv Kumar, Mehryar Mohri, and Ameet Talwalkar. Sampling techniques for the Nyström method. In *Proceedings of the 12th International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 304–311, 2009.

[KMTS16] Varun Kanade, Frederik Mallmann-Trenn, and Thomas Sauerwald. On coalescence time in graphs–when is coalescing as fast as meeting? *arXiv:1611.02460*, 2016.

[KO87] Hermann Kopetz and Wilhelm Ochsenreiter. Clock synchronization in distributed real-time systems. *IEEE Transactions on Computers*, 100(8):933–940, 1987.

[Koi96] Pascal Koiran. VC dimension in circuit complexity. In *Proceedings of the Eleventh Annual IEEE Conference on Computational Complexity*, pages 81–85. IEEE, 1996.

[KS70] N. I. Kokovkin-Shcherbak. Minimization of numerical algorithms for solving arbitrary systems of linear equations. 22(4):494-502, 1970.

[KS16] Rasmus Kyng and Sushant Sachdeva. Approximate Gaussian elimination for Laplacians: Fast, sparse, and simple. In *Proceedings of the 57th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, 2016.

[KSV08] Ravindran Kannan, Hadi Salmasian, and Santosh Vempala. The spectral method for general mixture models. *SIAM Journal on Computing*, 38(3):1141–1156, 2008.

[KU87] Christof Koch and Shimon Ullman. Shifts in selective visual attention: towards the underlying neural circuitry. In *Matters of Intelligence*, pages 115–141. Springer, 1987.

[KW92] Jacek Kuczyński and Henryk Woźniakowski. Estimating the largest eigenvalue by the power and Lanczos algorithms with a random start. *SIAM Journal on Matrix Analysis and Applications*, 13(4):1094–1122, 1992.

[LACBL16] Haipeng Luo, Alekh Agarwal, Nicolo Cesa-Bianchi, and John Langford. Efficient second order online learning by sketching. In *Advances in Neural Information Processing Systems 29 (NIPS)*, pages 902–910, 2016.

[LB07] Luisa Lima and Joao Barros. Random walks on sensor networks. In *5th International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks and Workshops*, pages 1–5. IEEE, 2007.

[LBH15] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[LBKW14] Yingyu Liang, Maria-Florina Balcan, Vandana Kanchanapally, and David P. Woodruff. Improved distributed principal component analysis. In *Advances in Neural Information Processing Systems 27 (NIPS)*, pages 3113–3121, 2014.

[LG12] François Le Gall. Faster algorithms for rectangular matrix multiplication. In *Proceedings of the 53rd Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 514–523, 2012.

[LG14] François Le Gall. Powers of tensors and fast matrix multiplication. In *Proceedings of the 39th International Symposium on Symbolic and Algebraic Computation*. ACM, 2014.

[LHLS17] Xuanqing Liu, Cho-Jui Hsieh, Jason D. Lee, and Yuekai Sun. An inexact subsampled proximal Newton-type method for large-scale machine learning. *arXiv:1708.08552*, 2017.

[LIKB99] Dale K Lee, Laurent Itti, Christof Koch, and Jochen Braun. Attention activates winner-take-all competition among visual filters. *Nature Neuroscience*, 2(4):375–381, 1999.

[LJS16] Chengtao Li, Stefanie Jegelka, and Suvrit Sra. Fast DPP sampling for Nyström with application to kernel methods. In *Proceedings of the 33rd International Conference on Machine Learning (ICML)*, pages 2061–2070, 2016.

[LKL10] Mu Li, James Tin-Yau Kwok, and Baoliang Lu. Making large-scale Nyström approximation possible. In *Proceedings of the 27th International Conference on Machine Learning (ICML)*, page 631, 2010.

[LL12] Jianguo Lu and Dingding Li. Sampling online social networks by random walk. In *Proceedings of the First ACM International Workshop on Hot Topics on Interdisciplinary Social Networks Research*, pages 33–40. ACM, 2012.

[LLNR14] Christoph Lenzen, Nancy Lynch, Calvin Newport, and Tsvetomira Radeva. Trade-offs between selection complexity and performance when searching the plane without communication. In *Proceedings of the 2014 ACM Symposium on Principles of Distributed Computing (PODC)*, pages 252–261, 2014.

[LM12] Chao Li and Gerome Miklau. Measuring the achievable error of query sets under differential privacy. *arXiv:1202.3399*, 2012.

[LMP13] Mu Li, Gary L. Miller, and Richard Peng. Iterative row sampling. In *Proceedings of the 54th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 127–136, 2013.

[LMP17a] Nancy Lynch, Cameron Musco, and Merav Parter. Computational tradeoffs in biological neural networks: Self-stabilizing winner-take-all networks. In *Proceedings of the 8th Conference on Innovations in Theoretical Computer Science (ITCS)*, 2017.

[LMP17b] Nancy Lynch, Cameron Musco, and Merav Parter. Neuro-RAM unit with applications to similarity testing and compression in spiking neural networks. In *Proceedings of the 31st International Symposium on Distributed Computing (DISC)*, 2017. Full version available at https://arxiv.org/abs/1706.01382.

[LMP17c] Nancy Lynch, Cameron Musco, and Merav Parter. Spiking neural networks: An algorithmic perspective. In *5th Workshop on Biological Distributed Algorithms (BDA 2017)*, July 2017.

[LMPV18] Robert Legenstein, Wolfgang Maass, Christos H Papadimitriou, and Santosh S Vempala. Long term memory and the densest k-subgraph problem. In *Proceedings of the 9th Conference on Innovations in Theoretical Computer Science (ITCS)*, 2018.

[LN15] Yu Lu and Sahand N Negahban. Individualized rank aggregation using nuclear norm regularization. In *53rd Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 1473–1479. IEEE, 2015.

[Lov93] László Lovász. Random walks on graphs: A survey. *Combinatorics, Paul Erdos is Eighty*, 2(1):1–46, 1993.

[LPDF08] Adi Livnat, Christos Papadimitriou, Jonathan Dushoff, and Marcus W Feldman. A mixability theory for the role of sex in evolution. *Proceedings of the National Academy of Sciences*, 105(50):19803–19808, 2008.

[LRMM88] John Lazzaro, Sylvie Ryckebusch, Misha Anne Mahowald, and Caver A Mead. Winner-take-all networks of o(n) complexity. Technical report, DTIC Document, 1988.

[LS14] Yin Tat Lee and Aaron Sidford. Path finding methods for linear programming: Solving linear programs in $\tilde{o}(\sqrt{rank})$ iterations and faster algorithms for maximum flow. In *Proceedings of the 55th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 424–433, 2014.

[LSS13] Quoc Le, Tamás Sarlós, and Alexander Smola. Fastfood - Computing Hilbert space expansions in loglinear time. In *Proceedings of the 30th*

International Conference on Machine Learning (ICML), pages 244–252, 2013.

[LSW09] Christoph Lenzen, Philipp Sommer, and Roger Wattenhofer. Optimal clock synchronization in networks. In *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems*, pages 225–238. ACM, 2009.

[LSW15] Yin Tat Lee, Aaron Sidford, and Sam Chiu-wai Wong. A faster cutting plane method and its implications for combinatorial and convex optimization. In *Proceedings of the 56th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 1049–1065, 2015.

[LW14] Jianguo Lu and Hao Wang. Variance reduction in large graph sampling. *Information Processing & Management*, 50(3):476–491, 2014.

[LYCG14] Lei Luo, Jian Yang, Jinhui Chen, and Yicheng Gao. Schatten p-norm based matrix regression model for image classification. In *Pattern Recognition*, pages 140–150. Springer, 2014.

[Maa96] Wolfgang Maass. On the computational power of noisy spiking neurons. In *Advances in Neural Information Processing Systems 9 (NIPS)*, pages 211–217, 1996.

[Maa97] Wolfgang Maass. Networks of spiking neurons: the third generation of neural network models. *Neural Networks*, 10(9):1659–1671, 1997.

[Maa99] Wolfgang Maass. Neural computation with winner-take-all as the only nonlinear operation. In *Advances in Neural Information Processing Systems 12 (NIPS)*, pages 293–299, 1999.

[Maa00] Wolfgang Maass. On the computational power of winner-take-all. *Neural Computation*, 12(11):2519–2535, 2000.

[Maa14] Wolfgang Maass. Noise as a resource for computation and learning in networks of spiking neurons. *Proceedings of the IEEE*, 102(5):860–880, 2014.

[McC84] Peter McCullagh. Generalized linear models. *European Journal of Operational Research*, 16(3):285–292, 1984.

[MD09] Michael W Mahoney and Petros Drineas. CUR matrix decompositions for improved data analysis. *Proceedings of the National Academy of Sciences*, 106(3):697–702, 2009.

[MEAM89] E Majani, Ruth Erlanson, and Yaser S Abu-Mostafa. On the k-winners-take-all network. In *Advances in Neural Information Processing Systems 2 (NIPS)*, pages 634–642, 1989.

[MM13] Xiangrui Meng and Michael W Mahoney. Low-distortion subspace embeddings in input-sparsity time and applications to robust linear regression. In *Proceedings of the 45th Annual ACM Symposium on Theory of Computing (STOC)*, 2013.

[MM15] Cameron Musco and Christopher Musco. Randomized block Krylov methods for stronger and faster approximate singular value decomposition. In *Advances in Neural Information Processing Systems 28 (NIPS)*, 2015.

[MM17] Cameron Musco and Christopher Musco. Recursive sampling for the Nyström method. In *Advances in Neural Information Processing Systems 30 (NIPS)*, 2017.

[MMG$^+$07] Alan Mislove, Massimiliano Marcon, Krishna P Gummadi, Peter Druschel, and Bobby Bhattacharjee. Measurement and analysis of online social networks. In *Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement*, pages 29–42, 2007.

[MNS$^+$18] Cameron Musco, Praneeth Netrapalli, Aaron Sidford, Shashanka Ubaru, and David P. Woodruff. Spectrum approximation beyond fast matrix multiplication: Algorithms and hardness. *Proceedings of the 9th Conference on Innovations in Theoretical Computer Science (ITCS)*, 2018.

[Mor73] Jacques Morgenstern. Note on a lower bound on the linear complexity of the fast Fourier transform. *Journal of the ACM (JACM)*, 20(2):305–306, 1973.

[MP69] Marvin Minsky and Seymour Papert. *Perceptrons.* MIT Press, 1969.

[MS99] Wolfgang Maass and Michael Schmitt. On the complexity of learning for spiking neurons with temporal coding. *Information and Computation*, 153(1):26–46, 1999.

[MSL17] Cameron Musco, Hsin-Hao Su, and Nancy Lynch. Ant-inspired density estimation via random walks. *Proceedings of the National Academy of Sciences*, 114(40):10534–10541, 2017. Preliminary version in the 2016 ACM Symposium on Principles of Distributed Computing (PODC), 2016.

[MSS91] Wolfgang Maass, Georg Schnitger, and Eduardo D Sontag. On the computational power of sigmoid versus boolean threshold circuits. In *Proceedings of the 32nd Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 767–776, 1991.

[MST15] Aleksander Madry, Damian Straszak, and Jakub Tarnawski. Fast generation of random spanning trees and the effective resistance metric. In *Proceedings of the 26th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2019–2036, 2015.

[MU05] Michael Mitzenmacher and Eli Upfal. *Probability and Computing: Randomized Algorithms and Probabilistic Analysis.* Cambridge University Press, 2005.

[Mus18] Christopher Musco. *Faster Linear Algebra for Data Analysis and Machine Learning.* PhD thesis, Massachusetts Institute of Technology, 2018.

[MW17a] Cameron Musco and David P. Woodruff. Is input sparsity time possible for kernel low-rank approximation? In *Advances in Neural Information Processing Systems 30 (NIPS)*, 2017.

[MW17b] Cameron Musco and David P. Woodruff. Sublinear time low-rank approximation of positive semidefinite matrices. In *Proceedings of the 58th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, 2017.

[MYK10] Abedelaziz Mohaisen, Aaram Yun, and Yongdae Kim. Measuring the mixing time of social graphs. In *Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement*, pages 383–389. ACM, 2010.

[NH10] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted Boltzmann machines. In *Proceedings of the 27th International Conference on Machine Learning (ICML)*, pages 807–814, 2010.

[NHD12] Feiping Nie, Heng Huang, and Chris Ding. Low-rank matrix recovery via efficient Schatten p-norm minimization. In *Twenty-Sixth AAAI Conference on Artificial Intelligence*, 2012.

[NN13] Jelani Nelson and Huy L Nguyên. OSNAP: Faster numerical linear algebra algorithms via sparser subspace embeddings. In *Proceedings of the 54th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, 2013.

[NNS$^+$14] Praneeth Netrapalli, UN Niranjan, Sujay Sanghavi, Animashree Anandkumar, and Prateek Jain. Non-convex robust PCA. In *Advances in Neural Information Processing Systems 27 (NIPS)*, pages 1107–1115, 2014.

[Now89] Steven J Nowlan. Maximum likelihood competitive learning. In *Advances in Neural Information Processing Systems 2 (NIPS)*, pages 574–582, 1989.

[NTD05] Stamatios C Nicolis, Guy Theraulaz, and Jean-Louis Deneubourg. The effect of aggregates on interaction rate in ant colonies. *Animal Behaviour*, 69(3):535–540, 2005.

[NWC$^+$12] Feiping Nie, Hua Wang, Xiao Cai, Heng Huang, and Chibiao Ding. Robust matrix completion via joint Schatten p-norm and l_p-norm minimization. In *IEEE 12th International Conference on Data Mining (ICDM)*, pages 566–574. IEEE, 2012.

[NWS02] Mark EJ Newman, Duncan J Watts, and Steven H Strogatz. Random graph models of social networks. *Proceedings of the National Academy of Sciences*, 99:2566–2572, 2002.

[ODL09] Matthias Oster, Rodney Douglas, and Shih-Chii Liu. Computation with spikes in a winner-take-all network. *Neural Computation*, 21(9):2437–2465, 2009.

[OF04] Bruno A Olshausen and David J Field. Sparse coding of sensory inputs. *Current Opinion in Neurobiology*, 14(4):481–487, 2004.

[OL06] Matthias Oster and Shih-Chii Liu. Spiking inputs to a winner-take-all network. *Advances in Neural Information Processing Systems 19 (NIPS)*, 2006.

[Osb79] Neville N Osborne. Is Dale's principle valid? *Trends in Neurosciences*, 2:73–75, 1979.

[Osb13] Neville N Osborne. *Dale's Principle and Communication Between Neurons*. Elsevier, 2013. Based on a Colloquium of the Neurochemical Group of the Biochemical Society, Held at Oxford University, July 1982.

[PLSZ17] Victor Y Pan, Qi Luan, John Svadlenka, and Liang Zhao. Superfast CUR matrix algorithms, their pre-processing and extensions. *arXiv:1710.07946*, 2017.

[Pra05] Stephen C Pratt. Quorum sensing by encounter rates in the ant *Temnothorax albipennis*. *Behavioral Ecology*, 16(2):488–496, 2005.

[PRTV00] Christos H. Papadimitriou, Prabhakar Raghavan, Hisao Tamaki, and Santosh Vempala. Latent semantic indexing: A probabilistic analysis. *Journal of Computer and System Sciences*, 61(2):217–235, 2000.

[Rad17] Tsvetomira Radeva. *A Symbiotic Perspective on Distributed Algorithms and Social Insects*. PhD thesis, Dissertation, Massachusetts Institute of Technology, 2017.

[Ras04] Carl Edward Rasmussen. Gaussian processes in machine learning. In *Advanced Lectures on Machine Learning*, pages 63–71. Springer, 2004.

[Raz03] Ran Raz. On the complexity of matrix product. *SIAM Journal on Computing*, 32(5):1356–1369, 2003.

[RB15] Lisa Roux and György Buzsáki. Tasks for inhibitory interneurons in intact brain circuits. *Neuropharmacology*, 88:10–23, 2015.

[RFLHL11] Bernardo Rudy, Gordon Fishell, SooHyun Lee, and Jens Hjerling-Leffler. Three groups of interneurons account for nearly 100% of neocortical GABAergic neurons. *Developmental Neurobiology*, 71(1):45–61, 2011.

[RGDA97] Alexa Riehle, Sonja Grün, Markus Diesmann, and Ad Aertsen. Spike synchronization and rate modulation differentially involved in motor cortical function. *Science*, 278(5345):1950–1953, 1997.

[RML17] Tsvetomira Radeva, Cameron Musco, and Nancy Lynch. New perspectives on algorithmic robustness inspired by ant colony house-hunting. In *5th Workshop on Biological Distributed Algorithms (BDA 2017)*, July 2017.

[RR07] Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *Advances in Neural Information Processing Systems 20 (NIPS)*, 2007.

[RS03] Ran Raz and Amir Shpilka. Lower bounds for matrix product in bounded depth circuits with arbitrary gates. *SIAM Journal on Computing*, 32(2):488–513, 2003.

[Saa03] Yousef Saad. *Iterative Methods for Sparse Linear Systems.* Society for Industrial and Applied Mathematics, second edition, 2003.

[Saa11] Yousef Saad. *Numerical Methods for Large Eigenvalue Problems: Revised Edition*, volume 66. SIAM, 2011.

[San89] Terence D Sanger. Optimal unsupervised learning in a single-layer linear feedforward neural network. *Neural Networks*, 2(6):459–473, 1989.

[Sar06] Támas Sarlós. Improved approximation algorithms for large matrices via random projections. In *Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 143–152, 2006.

[Sax09] Nitin Saxena. Progress on polynomial identity testing. *Bulletin of the EATCS*, 99:49–79, 2009.

[Sha15] Ohad Shamir. A stochastic PCA and SVD algorithm with an exponential convergence rate. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, pages 144–152, 2015.

[She13] Jonah Sherman. Nearly maximum flows in nearly linear time. In *Proceedings of the 54th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 263–269, 2013.

[SHG06] Robert J Schafer, Susan Holmes, and Deborah M Gordon. Forager activation and food availability in harvester ants. *Animal Behaviour*, 71(4):815–822, 2006.

[Shp03] Amir Shpilka. Lower bounds for matrix product. *SIAM Journal on Computing*, 32(5):1185–1200, 2003.

[SKP17] Michael L Smith, Phoebe A Koenig, and Jacob M Peters. The cues of colony size: how honey bees sense that their colony is large enough to begin to invest in reproduction. *Journal of Experimental Biology*, 220(9):1597–1605, 2017.

[SLO13] Andreas Stathopoulos, Jesse Laeuchli, and Kostas Orginos. Hierarchical probing for estimating the trace of the matrix inverse on toroidal lattices. *SIAM Journal on Scientific Computing*, 35(5):S299–S322, 2013.

[SN94] Michael N Shadlen and William T Newsome. Noise, neural codes and cortical organization. *Current Opinion in Neurobiology*, 4(4):569–579, 1994.

[SS95] Hava T Siegelmann and Eduardo D Sontag. On the computational power of neural nets. *Journal of Computer and System Sciences*, 50(1):132–150, 1995.

[SS00] Alex J Smola and Bernhard Schökopf. Sparse greedy matrix approximation for machine learning. In *Proceedings of the 17th International Conference on Machine Learning (ICML)*, pages 911–918, 2000.

[SS02] Bernhard Schölkopf and Alexander J Smola. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT Press, 2002.

[SS08] Daniel A. Spielman and Nikhil Srivastava. Graph sparsification by effective resistances. In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing (STOC)*, pages 563–568, 2008.

[ST04] Daniel A Spielman and Shang-Hua Teng. Nearly-linear time algorithms for graph partitioning, graph sparsification, and solving linear systems. In *Proceedings of the 36th Annual ACM Symposium on Theory of Computing (STOC)*, pages 81–90, 2004.

[STC04] John Shawe-Taylor and Nello Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004.

[Stu] Robert Stufflebeam. Introduction to neurons, synapses, action potentials, and neurotransmission. In *The Mind Project*. http://www.mind.ilstu.edu/curriculum/modOverview.php?modGUI=232.

[SWZ16] Zhao Song, David P. Woodruff, and Huan Zhang. Sublinear time orthogonal tensor decomposition. In *Advances in Neural Information Processing Systems 29 (NIPS)*, 2016.

[SWZ17] Zhao Song, David P Woodruff, and Peilin Zhong. Low rank approximation with entrywise $l1$-norm error. In *Proceedings of the 49th Annual ACM Symposium on Theory of Computing (STOC)*, pages 688–701. ACM, 2017.

[SY05] Anthony Man-Cho So and Yinyu Ye. Theory of semidefinite programming for sensor network localization. In *Proceedings of the 16th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 405–414, 2005.

[TB97] Lloyd N. Trefethen and David Bau. *Numerical Linear Algebra*. SIAM, 1997.

[TB17] Daniel Ting and Eric Brochu. Optimal sub-sampling with influence functions. *arXiv:1709.01716*, 2017.

[Tho90] Simon J Thorpe. Spike arrival times: A highly efficient coding scheme for neural networks. *Parallel Processing in Neural Systems*, pages 91–94, 1990.

[Tra09] Thomas Trappenberg. *Fundamentals of Computational Neuroscience*. OUP Oxford, 2009.

[Tro15] Joel A Tropp. An introduction to matrix concentration inequalities. *arXiv:1501.01571*, 2015.

[TYUC16] Joel A Tropp, Alp Yurtsever, Madeleine Udell, and Volkan Cevher. Randomized single-view algorithms for low-rank matrix approximation. *arXiv:1609.00048*, 2016.

[US16] Shashanka Ubaru and Yousef Saad. Fast methods for estimating the numerical rank of large matrices. In *Proceedings of the 33rd International Conference on Machine Learning (ICML)*, pages 468–477, 2016.

[Val00a] Leslie G Valiant. *Circuits of the Mind*. Oxford University Press, 2000.

[Val00b] Leslie G Valiant. A neuroidal architecture for cognitive computation. *Journal of the ACM (JACM)*, 47(5):854–882, 2000.

[Val05] Leslie G Valiant. Memorization and association on a realistic neural model. *Neural cCmputation*, 17(3):527–555, 2005.

[Vap98] Vladimir Vapnik. *Statistical Learning Theory*, volume 1. Wiley New York, 1998.

[VGLH06] Roy Varshavsky, Assaf Gottlieb, Michal Linial, and David Horn. Novel unsupervised feature filtering of biological data. *Bioinformatics*, 22(14):e507–e513, 2006.

[VVAE94] Carl Van Vreeswijk, LF Abbott, and G Bard Ermentrout. When inhibition not excitation synchronizes neural firing. *Journal of Computational Neuroscience*, 1(4):313–321, 1994.

[Wai15] Martin J Wainwright. High-dimensional statistics: A non-asymptotic viewpoint, draft. http://www.stat.berkeley.edu/~mjwain/stat210b/Chap2_TailBounds_Jan22_2015.pdf, 2015.

[Wei03] Eric W Weisstein. Gershgorin circle theorem. *MathWorld: A Wolfram Web Resource*, 2003.

[Wil12] Virginia Vassilevska Williams. Multiplying matrices faster than Coppersmith-Winograd. In *Proceedings of the 44th Annual ACM Symposium on Theory of Computing (STOC)*, pages 887–898, 2012.

[Wil15] Virginia Vassilevska Williams. Hardness of easy problems: Basing hardness on popular conjectures such as the strong exponential time hypothesis (invited talk). In *10th International Symposium on Parameterized and Exact Computation, IPEC 2015, September 16-18, 2015, Patras, Greece*, pages 17–29, 2015.

[Win70] Shmuel Winograd. On the number of multiplications necessary to computer certain functions. 23:165-179, 1970.

[Win87] Shmuel Winograd. Arithmetic complexity of computations. 1987.

[WLZ16] Shusen Wang, Luo Luo, and Zhihua Zhang. SPSD matrix approximation via column selection: theories, algorithms, and extensions. *Journal of Machine Learning Research*, 17(49):1–49, 2016.

[WMK$^+$02] Masahito Watanabe, Kentaro Maemura, Kiyoto Kanbara, Takumi Tamayama, and Hana Hayasaki. GABA and GABA receptors in the central nervous system and other organs. In *International Review of Cytology*, volume 213, pages 1–47. Elsevier, 2002.

[Woo14] David P. Woodruff. Sketching as a tool for numerical linear algebra. *Foundations and Trends in Theoretical Computer Science*, 10(1-2):1–157, 2014.

[WR94] Steve R Waterhouse and Anthony J Robinson. Classification using hierarchical mixtures of experts. In *Proceedings of the IEEE Workshop Neural Networks for Signal Processing*, pages 177–186, 1994.

[WS01] Christopher Williams and Matthias Seeger. Using the Nyström method to speed up kernel machines. In *Advances in Neural Information Processing Systems 14 (NIPS)*, 2001.

[WS03] Wei Wang and Jean-Jacques E Slotine. k-winners-take-all computation with neural oscillators. *arXiv:q-bio/0401001*, 2003.

[WSO$^+$07] Thilo Womelsdorf, Jan-Mathijs Schoffelen, Robert Oostenveld, Wolf Singer, Robert Desimone, Andreas K Engel, and Pascal Fries. Modulation of neuronal interactions through neuronal synchronization. *Science*, 316(5831):1609–1612, 2007.

[WW10] Virginia Vassilevska Williams and Ryan Williams. Subcubic equivalences between path, matrix and triangle problems. In *Proceedings of the 51st Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, 2010.

[WZ13] Shusen Wang and Zhihua Zhang. Improving CUR matrix decomposition and the Nyström approximation via adaptive sampling. *Journal of Machine Learning Research*, 14:2729–2769, 2013.

[WZZ14] Shusen Wang, Tong Zhang, and Zhihua Zhang. Adjusting leverage scores by row weighting: A practical approach to coherent matrix completion. *arXiv:1412.7938*, 2014.

[XGL$^+$16] Yuan Xie, Shuhang Gu, Yan Liu, Wangmeng Zuo, Wensheng Zhang, and Lei Zhang. Weighted Schatten p-norm minimization for image denoising and background subtraction. *IEEE Transactions on Image Processing*, 25(10):4842–4857, 2016.

[XQT$^+$16] Y. Xie, Y. Qu, D. Tao, W. Wu, Q. Yuan, and W. Zhang. Hyperspectral image restoration via iteratively regularized weighted Schatten p-norm minimization. *IEEE Transactions on Geoscience and Remote Sensing*, PP(99):1–18, 2016.

[Yao77] Andrew Chi-Chin Yao. Probabilistic computations: Toward a unified measure of complexity. In *Proceedings of the 18th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 222–227, 1977.

[YG89] Alan L Yuille and Norberto M Grzywacz. A winner-take-all mechanism based on presynaptic inhibition feedback. *Neural Computation*, 1(3):334–347, 1989.

[YH38] Gale Young and Alston S Householder. Discussion of a set of points in terms of their mutual distances. *Psychometrika*, 3(1):19–22, 1938.

[YS07] Stephen J Young and Edward R Scheinerman. Random dot product graph models for social networks. In *International Workshop on Algorithms and Models for the Web-Graph*, pages 138–149. Springer, 2007.

[ZP96] Anthony M Zador and Barak A Pearlmutter. VC dimension of an integrate-and-fire neuron model. In *Proceedings of the 9th Annual Conference on Computational Learning Theory (COLT)*, pages 10–18. ACM, 1996.

[ZTK08] Kai Zhang, Ivor W. Tsang, and James T. Kwok. Improved Nyström low-rank approximation and error analysis. In *Proceedings of the 25th International Conference on Machine Learning (ICML)*, pages 1232–1239, 2008.