Proceedings of Machine Learning Research vol 178:1–60, 2022 35th Annual Conference on Learning Theory

Hierarchical Clustering in Graph Streams: Single-Pass Algorithms and Space Lower Bounds

Sepehr Assadi SEPEHR.ASSADI@RUTGERS.EDU
Rutgers University

Vaggos Chatziafratis VAGGOS@STANFORD.EDU
University of California – Santa Cruz (UCSC)

Jakub Łącki JLACKI@GOOGLE.COM
Google Research

Vahab Mirrokni MIRROKNI@GOOGLE.COM
Google Research

Chen Wang CHEN.WANG.CS@RUTGERS.EDU
Rutgers University

Editors: Po-Ling Loh and Maxim Raginsky

Abstract

The Hierarchical Clustering (HC) problem consists of building a hierarchy of clusters to represent a given dataset. Motivated by the modern large-scale applications, we study the problem in the *streaming model*, in which the memory is heavily limited and only a single or very few passes over the input are allowed. Specifically, we investigate whether a good hierarchical clustering can be obtained, or at least whether we can approximately estimate the value of the optimal hierarchy. To measure the quality of a hierarchy, we use the HC minimization objective introduced by Dasgupta Dasgupta (2016). Assuming that the input is an n-vertex weighted graph whose edges arrive in a stream, we derive the following results on space-vs-accuracy tradeoffs:

- With $O(n \cdot \text{polylog}\, n)$ space, we develop a single-pass algorithm, whose approximation ratio matches the currently best *offline* algorithm Charikar and Chatziafratis (2017).
- When the space is more limited, namely, $n^{1-o(1)}$, we prove that no algorithm can even estimate the value of the optimum hierarchical tree to within an $o(\frac{\log n}{\log \log n})$ factor, even when allowed polylog n passes over the input and exponential time.
- In the most stringent setting of polylog n space, studied extensively in the literature, we rule out algorithms that can even distinguish between "highly"-vs-"poorly" clusterable graphs, namely, graphs that have an $n^{1/2-o(1)}$ factor gap between their HC objective value.
- Finally, we prove that any single-pass streaming algorithm that computes an optimal HC clustering requires storing almost the entire input even if allowed exponential time.

Our algorithmic results establish a general structural result that proves that cut sparsifiers of input graphs can preserve the cost of "balanced" hierarchical trees to within a constant factor, and thus can be used in place of the original (dense) graphs when solving HC. Our lower bound results include a new streaming lower bound for a novel problem "One-vs-Many-Expanders", which can be of independent interest.

Keywords: Hierarchical Clustering, Streaming Algorithms, Sublinear Memory, Dasgupta's Objective, Lower Bounds, Communication Complexity

1. Introduction

Motivated by a variety of data mining and computational biology applications, Hierarchical Clustering (HC) is the canonical problem of building a hierarchy of clusters to represent a dataset. This hierarchy takes the form of a rooted binary tree (also called a "dendrogram") whose leaves are in one-to-one correspondence with the data points, thus capturing their relationships at various levels of granularity. Representing a dataset as a tree structure offers several advantages: there is no need to specify the number of clusters in advance, HC is easy to interpret and visualize, and there are simple-to-implement HC algorithms available (e.g., either top down divisive or bottom up linkage methods). As a result, HC has played a prominent role both in theory and in practice across different domains, with canonical applications ranging from biology and statistics to finance and sociology (Cavalli-Sforza and Edwards, 1967; Berkhin, 2006; Eisen et al., 1998; Felsenstein, 2004; Hastie et al., 2009; Tumminello et al., 2010; Bateni et al., 2017a; Mann et al., 2008).

Deploying HC algorithms in practice however is a challenging task. In particular, a major challenge is achieving good scalability. With the rise of data-intensive applications, there is dire need to solve HC for extremely large datasets. Additionally, these datasets are typically evolving over time (e.g., new queries/users/videos added in a platform), thus making said scaling issues even harder to deal with. The best known algorithms for some commonly used linkage methods, such as Average Linkage, suffer from quadratic runtime (in the number of data points) which is prohibitive in modern settings. To overcome these issues, recent efforts have focused on accelerating bottom-up linkage methods (Loewenstein et al., 2008; Abboud et al., 2019; Bateni et al., 2017b; Monath et al., 2019a, 2021; Sumengen et al., 2021; Dhulipala et al., 2021) or top-down divisive methods (Avdiukhin et al., 2019) and on exploiting geometric embedding techniques (Naumov et al., 2021; Rajagopalan et al., 2021; Nickel and Kiela, 2017).

In this paper, we study HC in the *graph streaming model*, which is a canonical model designed to capture the essence of large-scale computation. Graph streaming algorithms process their input by making one (or few) sequential pass(es) over their edges while using a limited memory, much smaller than the input size. These constraints capture several challenges of processing massive graphs such as I/O-efficiency or monitoring evolving graphs; see, e.g. (Muthukrishnan, 2005; Feigenbaum et al., 2005; McGregor, 2014) and references therein. The main motivation behind our work is the following question:

If we are allowed only a single sequential pass over the data and a limited space, how good a hierarchical clustering can we compute? In general, what are the space-vs-*accuracy tradeoffs?*

We present several algorithmic and impossibility results that address this question. On the algorithmic front, we design a single-pass HC algorithm minimizing Dasgupta's HC cost function (Dasgupta, 2016), that matches the guarantees of known non-streaming algorithms (Charikar and Chatziafratis, 2017; Cohen-Addad et al., 2019), while using memory proportional to the number of data points (and thus quadratically smaller than the input size that contains pairwise similarities of the data points). On the lower bounds front, we give several impossibility results across a range of various (sublinear) memory regimes, providing tradeoffs for the space required in order to obtain "good" HC trees or to estimate their values, as measured by Dasgupta's objective (Dasgupta, 2016). We elaborate more on our results in Section 1.2.

To the best of our knowledge, we are the first to provide theoretical guarantees for streaming HC under Dasgupta's cost function in the general graph similarity setting (i.e., the input need not satisfy triangle inequality), and/or under memory limitations or single-pass/few-pass desiderata. In contrast, recent results in (Rajagopalan et al., 2021) hold only for metric data in $\mathbb{R}^d$ and their focus is on maximization HC objectives (Moseley and Wang, 2017; Cohen-Addad et al., 2019) (which are provably shown to be easier to approximate (Charikar et al., 2019; Alon et al., 2020; Naumov et al., 2021)).

1.1. Background, Problem Definition, and Related Work

Before stating our results in more detail, we start with a brief description of prior work in the literature of *optimization-based* hierarchical clustering. The main motivating question here is "how does one evaluate the quality of a hierarchical tree on a given dataset?".

Despite its popularity and importance, HC is underdeveloped from a theoretical perspective. In particular, many heuristics for HC are defined procedurally rather than in terms of an optimization objective; as such they lack theoretical analyses on their performance guarantees. Indeed, until recently, there was no global objective function for HC to evaluate how good or bad a proposed solution is, in stark contrast with the multitude of objectives we typically encounter in "flat" clustering (e.g., k-means, k-medians, k-multicut, correlation clustering, etc.). Having an appropriate objective allows us to evaluate the performance of different algorithms, to quantify their success or failures, and in some cases, to add explicit constraints for the hierarchy (Kleindessner and von Luxburg, 2017; Vikram and Dasgupta, 2016; Chatziafratis et al., 2018), similar to "must-link/cannot-link" constraints in k-means (Wagstaff and Cardie, 2000; Wagstaff et al., 2001).

In an influential work, Dasgupta (2016) proposed a minimization objective for HC based on pairwise similarity information on n data points. Under this objective, the data is embedded as a graph $G = (V, E, w)$, where the vertices are the data points, and edges are obtained by pairwise similarity. The clustering is represented by a rooted tree $\mathcal{T}$, where each leaf node contains a single vertex, and each non-leaf node of $\mathcal{T}$ induces a cluster (as such, the root contains V). The inclusion of sub-clusters is characterized by the clusters induced by child nodes. The total cost is measured by the summation of the (weighted) pairwise costs, where the cost between vertex pair (u, v) is defined as the (weighted) number of leaf nodes induced by the subtree rooted at the lowest common ancestor between u and v. More formally, the problem definition can be given as follows.

Problem 1 (HC under Dasgupta's cost function) *Given an n-vertex weighted graph $G = (V, E, w)$ with vertices corresponding to data points and edges measuring their similarity, create a rooted tree $\mathcal{T}$ whose leaf-nodes are V. The goal is to* minimize *the cost of this tree $\mathcal{T}$ defined as*

$$\mathbf{cost}_G(\mathcal{T}) := \sum_{e=(u,v)\in E} w(e) \cdot |\text{leaf-nodes}(\mathcal{T}[u \vee v])|, \tag{1}$$

where $u \vee v$ is the node that is the lowest common ancestor of u and v, $\mathcal{T}[u \vee v]$ is the sub-tree rooted from $u \vee v$, and $|leaf\text{-}nodes(\mathcal{T}[u \vee v])|$ is the number of leaf-nodes in the sub-tree. We use $\mathsf{OPT}(G)$ *to denote the cost of an optimal tree for the graph G.*

Dasgupta (2016) gave a poly-time $O(\alpha(n) \cdot \log n)$-approximation algorithm for the aforementioned hierarchical clustering problem, where $\alpha(n)$ denotes the best approximation ratio possible for the *Sparsest Cut* problem (currently, $\alpha(n) = O(\sqrt{\log n})$ (Arora et al., 2009)). Follow-up works improved on this result by proving an $O(\log n)$ approximation via linear programming (Roy and Pokutta, 2016) and an $O(\sqrt{\log n})$ approximation via semidefinite programming (Charikar and Chatziafratis, 2017). In addition, (Charikar and Chatziafratis, 2017; Cohen-Addad et al., 2019) improved the analysis of Dasgupta (2016) based on sparsest cut problem to achieve an $O(\alpha(n))$-approximation (Charikar and Chatziafratis (2017) also provides a similar algorithm using *Balanced Cut* as a subroutine instead of sparsest cut). On the hardness front, Charikar and Chatziafratis (2017) proved that under the *Small Set Expansion (SSE) Hypothesis*, there is no constant factor approximation algorithm for Dasgupta's hierarchical clustering problem in polynomial time.

More generally, Dasgupta's objective has led to a flurry of both theoretical and empirical results about the computational complexity and optimization of HC, expanding our understanding of HC and mirroring the important progress made in the "flat" clustering literature over the past several decades. Such results include approximation guarantees for old linkage algorithms (Moseley and Wang, 2017; Cohen-Addad et al., 2019), explaining success of existing methods (Charikar et al., 2019; Avdiukhin et al., 2019), designing novel approaches to HC (Roy and Pokutta, 2016; Chatziafratis et al., 2018), characterizing its computational complexity and inapproximability (Charikar and Chatziafratis, 2017; Chatziafratis et al., 2021), and novel connections to hyperbolic embeddings (Chami et al., 2020; Monath et al., 2019b).

In this work, we study Dasgupta's hierarchical clustering problem in the graph streaming model, wherein the edges of the input graph G are arriving one by one in an arbitrary order, and the algorithm is allowed to make a single pass over these edges and compute an HC tree $\mathcal{T}$ that (approximately) minimizes $\mathbf{cost}_G(\mathcal{T})$ (or estimate $\mathbf{cost}_G(\mathcal{T})$ studied in some of our lower bounds).

1.2. Our Contributions

We provide a comprehensive treatment of HC in the graph streaming model. Our first result gives an algorithm for obtaining an approximation ratio proportional to the best non-streaming algorithm, while using only $\widetilde{O}(n) := O(n \cdot \text{polylog}(n))$[1] space, referred to as *Semi-Streaming* space restriction (Feigenbaum et al., 2005), the so-called 'sweet spot' for graph streaming algorithms.

> **Result 1** *There exists a single-pass streaming algorithm for hierarchical clustering (Theorem 1) that uses $O(n \cdot \text{polylog}\, n)$ space and achieves an $O(\sqrt{\log n})$-approximation in polynomial time or $O(1)$-approximation in exponential time.*

Result 1 gives us the best of both worlds: approximation ratio asymptotically matching best non-streaming algorithm of Charikar and Chatziafratis (2017) (by using it in a black-box way) and space complexity that is only larger than the output clustering by poly $\log n$ factors. Moreover, as we describe later, this result can use many other HC algorithms or heuristics as a black-box in place of Charikar and Chatziafratis (2017) (e.g., to gain faster runtime), while achieving asymptotically the same approximation ratio as the black-box algorithm.

As we shall explain more in Section 1.3, our Result 1 is based on a general *sparsification* approach that can be used in a variety of other settings as well. For instance, it also implies an $O(1)$-round Massively Parallel Computation (MPC) algorithm for HC on machines of memory $\widetilde{O}(n)$; see, e.g. (Karloff et al., 2010; Ahn et al., 2012; Beame et al., 2013; Kumar et al., 2013; Czumaj et al., 2018; Assadi et al., 2019b,a) and references therein for more details on the MPC model and its connection to streaming, among others.

1. Throughout, we use $\widetilde{O}(\cdot)$ and $\widetilde{\Omega}(\cdot)$ notation to suppress poly $\log(n)$ factors.

The space complexity of our algorithm in Result 1 is nearly optimal as any streaming algorithm that outputs a clustering of input points requires $\Omega(n \log n)$ bits of space just to store the answer. However, in many scenarios, one is interested in algorithms that can *distinguish* between "highly clusterable" inputs versus "poorly clusterable" ones; in other words, be able to only *estimate* the cost of the best HC tree in Dasgupta's hierarchical clustering problem (see, e.g. (Kapralov et al., 2015; Kapralov and Krachun, 2019; Guruswami and Tao, 2019; Assadi et al., 2020; Chou et al., 2020; Assadi and N, 2021) for a vibrant area of research on these streaming estimation problems for property testing or constraint satisfaction problems). While our algorithm in Result 1 clearly also works for the estimation problem, its space can no longer be considered nearly-optimal a priori. Our next result addresses this.

Result 2 *Any streaming algorithm with a memory of $n^{1-o(1)}$ cannot estimate the hierarchical clustering objective value to within an $o(\frac{\log n}{\log\log n})$ factor even if allowed* $\text{polylog}(n)$ *passes and exponential time.*

This result effectively rules out any algorithm with $n^{1-o(1)}$ memory to achieve an approximation ratio as competitive as Result 1 for the estimation problem (even when allowed exponential time and an "unreasonably large" number of passes). It is worth noting that while we obtain this result by a reduction from streaming lower bounds of Assadi and N (2021), this is the first application of these techniques to proving lower bounds for $\omega(1)$ approximation factors as well as $\omega(\log n)$ passes.

While quite strong in terms of space (and passes), Result 2 still leaves out possibility of algorithms with approximation ratio of $O(\log n)$, which are quite acceptable for HC. On the other end of the spectrum, one can ask how well of an approximation can we hope for on the most stringent restriction of $\text{polylog}(n)$ space and one pass? (this is the setting most focused on in "classical" streaming literature starting from Alon et al. (1996), as well as aforementioned line of work on estimation problems in graph streams). Our next result suggests that the answer is *"not much"*.

Result 3 *Any single-pass streaming algorithm with* $\text{polylog}(n)$ *space cannot estimate the hierarchical clustering value with an approximation ratio of $n^{1/2-\delta}$ for any constant $\delta > 0$.*

Proof of Result 3 turned out to be the most technically challenging part of our paper, as we can no longer rely on reductions from existing streaming lower bounds. En route to proving this result, we establish a general streaming lower bound: no $\text{polylog}(n)$ space streaming algorithm can distinguish between inputs consisting of a single *expander* versus a collection of many *small* vertex-disjoint expanders. This problem is the "expander-variant" of the by-now famous *gap cycle counting* problem of Verbin and Yu (2011) (where instead of expanders, we have *cycles* in the input) that has found numerous applications in streaming lower bounds (including our Result 2); see, e.g. (Verbin and Yu, 2011; Kapralov et al., 2015; Assadi et al., 2020; Assadi and N, 2021; Kapralov et al., 2022) and references therein. Our expander-variant of this problem seems versatile enough to find other applications and is therefore interesting in its own right.

Finally, going back to Result 1 and the $\widetilde{O}(n)$-space regime, we can ask whether settling for approximation was even necessary for this problem. In particular, can we match the performance of best non-streaming algorithms *exactly* (not asymptotically) or better yet obtain an exact optimal solution in exponential time? Our final result rules out this possibility also as long as the space of our algorithm is less than the input size (at which point, we can trivially store the entire input and solve the problem offline in exponential time).

Result 4 *Any single-pass streaming algorithm for finding an optimal hierarchical clustering tree (or even determining its cost) requires a memory of $\Omega(n^2)$ bits.*

This concludes the description of our main results. Putting these results together, our paper has the following message. It is possible to solve HC with asymptotically the same approximation ratio as that of best known non-streaming algorithms, while using only $\widetilde{O}(n)$ space (Result 1). But, reducing the space to $n^{1-o(1)}$ prohibits us from getting competitive approximations even in polylog(n) passes (Result 2), and reducing the space further to $n^{o(1)}$ prohibits us from even distinguishing between inputs with $n^{1/2-o(1)}$ factor gap between their optimal HC cost (Result 3). Finally, even increasing the space to $o(n^2)$ is not going to remove the need for approximation (Result 4).

1.3. Our Techniques

Algorithmic results. Dasgupta's work (Dasgupta, 2016) on introducing the objective cost for HC resulted in beautiful connections between HC and standard cut-based graph problems such as sparsest cut and balanced cut. For instance, the algorithm proposed by Dasgupta (2016) for HC is to recursively partition the vertices of the graph across an approximate sparsest cut at each level of the hierarchical tree until we reach the leaf-nodes. The work of Dasgupta (2016) shows that approximation ratio of this algorithm is $O(\alpha(n) \cdot \log n)$ where $\alpha(n)$ is the approximation ratio of the black-box sparsest cut algorithm we use, and follow up works in (Charikar and Chatziafratis, 2017; Cohen-Addad et al., 2019) improved the analysis to an $O(\alpha(n))$ approximation.

When it comes to graph streaming, many of cut-based problems including sparsest cut and balanced cut have a standard solution using *Cut Sparsifiers* (Benczúr and Karger, 2015): these are (re-weighted) subgraphs of the input graph that preserve the value of every *global* cut approximately, while being quite sparse with only $\widetilde{O}(n)$ edges. By now, there are simple streaming algorithms for recovering cut sparsifiers in $\widetilde{O}(n)$ space (see, e.g. (McGregor, 2014)), and it is easy to see that running a non-streaming algorithm for the cut-based problem on this sparsifier, results also in solutions of approximately the same quality on the original graph. Yet, this recipe does *not* apply to HC: in the aforementioned connection of sparsest cut and HC, one needs to solve sparsest cut recursively on *induced* subgraphs of the input after the first level of recursion – this in turn requires our cut sparsifier to not only preserve global cuts but also induced cuts, i.e., the weight of edges between any two subsets (A, B) of vertices (not only $(A, \bar{A})$). It is easy to see that such a "sparsifier" requires to store all edges of the graph!

Our main algorithmic contribution in this paper is to bypass this challenge. Instead of considering each separate (induced) cut that may appear when running standard HC algorithms such as (Dasgupta, 2016; Charikar and Chatziafratis, 2017), we prove a "global" structural property of cut sparsifiers for HC directly: the HC cost of any *balanced* hierarchical tree[2] as a whole remains almost the same between the original graph and its cut sparsifier (even though costs of some sub-trees can deviate dramatically). Given that the HC algorithm of Charikar and Chatziafratis (2017) also optimizes only over balanced hierarchical trees, we obtain that running that algorithm over the sparsifier, instead of the entire graph, will result in a solution with only a constant factor worse approximation guarantee. This way, we get a general recipe for solving HC using cut sparsifiers also which is applicable to graph streaming among other models such as MPC mentioned earlier (we

2. By a balanced tree, we mean a tree where at every node, the size of sub-trees of each child-node is within a constant factor of the other ones. See Theorem 17 for the formal definition.

further show that any HC problem admits an $O(1)$-approximation balanced solution, so restricting ourselves to balanced solutions is never going to cost us much).

Lower bound results. The starting point of our lower bound in Result 2 is the streaming lower bound for the (noisy) gap cycle counting problem of Assadi and N (2021)[3]. Informally speaking, Assadi and N (2021) proved that any polylog(n) pass algorithm that can distinguish between graphs composed of vertex-disjoint cycles of length $\Theta(n)$ or vertex-disjoint cycles of length polylog(n) requires $n^{1-o(1)}$ space (the actual problem definition involves also some "noisy" paths; see Section 3). Using the result of Dasgupta (2016) that characterizes the HC cost of vertex-disjoint graphs as well as cycles, one can show that the cost optimal hierarchical tree differs by a factor of $\Theta(\frac{\log n}{\log \log n})$ between these two family of graphs, which implies our desired lower bound as well via a reduction to Assadi and N (2021).

Our Result 3 is considerably more involved and is our main contribution on the lower bound front. The main challenge is that to prove a strong approximation lower bound, we can no longer rely on using "loosely-connected" graphs such as cycles (as their optimal HC cost is not going to be that different between the two cases). Because of this, we introduce the *One vs. Many Expanders (OvME)* problem wherein the goal is to distinguish between graphs consisted of a single expander with $\Theta(\log n)$-degree and $\Theta(\log n)$-edge expansion (see Theorem 21), versus $n^{1/2+o(1)}$ vertex-disjoint expanders with the same guarantees. A simple argument, using properties of expanders, allows to bound the difference in the optimal HC cost between these two families with an $n^{1/2-o(1)}$ factor. The bulk of our effort is then to prove the lower bound for this family of input graphs which are inherently different from cycles[4]. On a (very) high level, the proof of this lower bound is by (i) designing a multi-party *communication game* in spirit of (Kapralov et al., 2015; Chou et al., 2020, 2021) and reducing it to a two-party one using a standard hybrid argument in (Kapralov et al., 2015), (ii) applying a *decorrelation step* to this game to reduce the problem to proving a low-probability-of-success lower bound in spirit of (Assadi and N, 2021), and (iii) using a Fourier analytic method originated in (Gavinsky et al., 2007) based on *KKL inequality* (Kahn et al., 1988), to establish the lower bound (our decorrelation step, based on a new notion of "advantage" of protocols using KL-divergence, is the one that greatly deviates from prior work in (Kapralov et al., 2015; Chou et al., 2020, 2021; Assadi and N, 2021) and allows us to use Fourier analytic tools to analyze our final problem, despite its considerable differences from prior problems).

Finally, Result 4 is established using a reduction from the *Index* communication game (Ablayev, 1993). We create a family of graphs consisting of $\Theta(n)$-vertex "near-cliques" with few edges between them so that the value of optimum HC cost depends on a single edge in the input graph, which cannot be detected by an $o(n^2)$-space streaming algorithm. Our proof of this part extends prior work of Dasgupta (2016) on characterizing optimal HC costs on paths and cliques, to slightly more complex graphs.

Recent Independent Work. Independently of our work, Agarwal et al. (2022) also studied HC under Dasgupta's cost function in the settings similar to our paper. Whereas our focus has been pri-

3. We note that we use a slight variation of the problem that follows immediately from Assadi and N (2021) but is somewhat different from the description in that work.
4. E.g., being expanders they are way-more well connected and have much shorter diameter; see (Kapralov et al., 2015; Assadi and N, 2021; Kapralov et al., 2022) for the role of these parameters in prior lower bounds. Note also that the result of (Kapralov et al., 2015; Kapralov and Krachun, 2019) can be seen as proving a lower bound for distinguishing between a single expander versus *two* expanders as opposed to $n^{1/2-o(1)}$ many in our work.

marily in the streaming setting and space complexity of algorithms, Agarwal et al. (2022) focused on designing sublinear time algorithms in the query model and sublinear communication algorithms in the MPC model. But, similar to our Result 1 (and Theorem 2 specifically), they also prove a general structural result that shows that a cut sparsifier can be used to recover a $(1+o(1))$-approximation to the underlying HC instance, which is stronger than our $O(1)$-approximation guarantee. As a result, they can also recover our Result 1 with improved leading constant in the approximation. This improvement also applies to the MPC model where they show that $\tilde{O}(n)$ memory per machine suffices to get a 2-round algorithm that achieves $(1+o(1))$-approximation (they prove that any *one*-round polylog(n)-approximation MPC algorithm requires $\Omega(n^{4/3-o(1)})$ memory per machine). Beside this algorithmic connection, the rest of our work and Agarwal et al. (2022) are entirely disjoint.

2. A Semi-Streaming Algorithm for Hierarchical Clustering

We introduce our main upper bound result in this section, which gives a single-pass streaming algorithm that uses a memory of $\tilde{O}(n)$ words and asymptotically matches the approximation factor of the best *offline* HC algorithms. As mentioned before, the high-level idea of our algorithm is to maintain a $(1\pm\varepsilon)$-cut sparsifier throughout the stream, and run offline HC algorithms on the sparsifier graph. Since Eq (2) gives a way of expressing the cost of $\mathcal{T}$ as sum of costs of a series of induced cuts, the cut sparsifier intuitively 'preserves' the quality of cut-based heuristic algorithms. However, the main roadblock for such an idea is that the cut sparsifier only (approximately) preserves the value of *global* cuts and *not* necessarily the *induced* ones (as in Eq (2)). In this section, we settle the problem in Section 2.1 by establishing the relationship between global cuts and the cost of the HC-trees. We then present the main algorithm in Section 2.2.

2.1. A Sparsification Result for Hierarchical Clustering

We now give the formal statement for the relationship between the costs of HC trees on G and on its $(1\pm\varepsilon)$-cut sparsifier H as follows.

Theorem 2 *Let $G=(V,E,w_G)$ be any weighted undirected graph, $H=(V,E_H,w_H)$ be an $(1\pm\varepsilon)$-cut sparsifier of G for some $\varepsilon\in(0,1)$, and $\mathcal{T}$ be any β-balanced HC-tree on vertices V. Then,*

$$(1-\varepsilon)\cdot\beta\cdot\mathbf{cost}_G(\mathcal{T})\leq\mathbf{cost}_H(\mathcal{T})\leq(1+\varepsilon)\cdot\frac{1}{\beta}\cdot\mathbf{cost}_G(\mathcal{T}).$$

The key step to prove Theorem 2 is the following lemma, which 'massages' the cost function in Eq (2) to a series of global cuts, albeit with some loss. This is done by crucially using the balanced property of the tree $\mathcal{T}$. On the high level, such a 'massage' is possible from balanced HC trees in the following sense. Suppose for every cut (A,B), instead of charging $w_G(A,B)$ with a $|A\cup B|$ multiplicative factor, let us additionally charge the edges in $w_{\text{extra}}(A,B):=w_G(A,\bar{A})\cup w_G(B,\bar{B})\setminus w_G(A,B)$ also with a $|A\cup B|$ multiplicative factor. Indeed, this introduces some extra terms to the cost. We will show that the extra costs introduced as such is at most a *constant* factor of the hierarchical clustering cost.

Fix a cut (A^*,B^*) and suppose it is associated with node u in $\mathcal{T}$. Note that by Eq (2), the edges in $w_G(A^*,B^*)$ *never* incur any costs outside the induced subtree $\mathcal{T}[u]$. Furthermore, for nodes inside the induced subtree $\mathcal{T}[u]$ (other than u itself), edges in $w_G(A^*,B^*)$ inccur costs by contributing to $w_{\text{extra}}(A, B)$, where either $A \cup B \subseteq A^*$ or $A \cup B \subseteq B^*$. Crucially, since $\mathcal{T}$ is *balanced*, the multiplicative factor $|A \cup B|$ on $e \in w_G(A^*, B^*)$ decreases *exponentially*. Therefore, the extra contribution for edges in $w_G(A^*, B^*)$ on the internal nodes of $\mathcal{T}[u]$ other than u follows a geometric series. As such, the overhead of the cost introduced by the global cut terms is at most an $O(1)$ multiplicative factor of the HC cost.

We now formalize the above intuition as the following lemma.

Lemma 3 *Let $G = (V, E, w)$ be any arbitrary graph and $\mathcal{T}$ be a β-balanced HC-tree of G. Define:*

$$W(G) := \sum_{\substack{(A,B) := \textbf{cut}(\mathcal{T}[u]) \text{ for} \\ \text{internal nodes } u \text{ of } \mathcal{T}}} \frac{1}{2} \cdot (w_G(A, \bar{A}) + w_G(B, \bar{B})) \cdot |A \cup B|$$

then,

$$\textbf{cost}_G(\mathcal{T}) \leq W(G) \leq \frac{1}{\beta} \cdot \textbf{cost}_G(\mathcal{T}).$$

Due to space limit, we defer the formal proof of Lemma 3 to Appendix C.1.

Proof of Theorem 2 Consider the values $W(G)$ and $W(H)$ as defined by Lemma 3 for graphs G and H, respectively. Since $W(\cdot)$ is a linear function of weights of global cuts and H is an ε-sparsifier of G, we have that,

$$(1 - \varepsilon) \cdot W(G) \leq W(H) \leq (1 + \varepsilon) \cdot W(G).$$

By applying Lemma 3 for $\textbf{cost}_G(\mathcal{T})$ and $\textbf{cost}_H(\mathcal{T})$, we have that,

$$\textbf{cost}_H(\mathcal{T}) \leq W(H) \leq (1 + \varepsilon) \cdot W(G) \leq \frac{1 + \varepsilon}{\beta} \cdot \textbf{cost}_G(\mathcal{T}),$$
$$\textbf{cost}_H(\mathcal{T}) \geq \beta \cdot W(H) \geq \beta \cdot (1 - \varepsilon) \cdot W(G) \geq \beta \cdot (1 - \varepsilon) \cdot \textbf{cost}_G(\mathcal{T}),$$

finalizing the proof. ■

2.2. A Semi-Streaming Algorithm for Hierarchical Clustering

Theorem 2 implies that a HC tree $\mathcal{T}$ that works well on the $(1 \pm \varepsilon)$-cut sparsifier H also performs well on G, provided $\mathcal{T}$ is balanced. Therefore, we can obtain an algorithm by first maintaining a $(1 \pm \varepsilon)$-cut sparsifier, and then finding a balanced HC tree with a good approximation factor on the sparsifier graph. This leads to our main algorithm, presented as follows.

Theorem 4 *There is a single-pass (deterministic) semi-streaming algorithm for hierarchical clustering that uses $O(n \log^3(n))$ space and achieves an $O(\sqrt{\log n})$-approximation in polynomial time and an $O(1)$ approximation in exponential time.*

Proof Throughout the stream, we simply maintain a $(1 \pm \varepsilon)$-cut sparsifier H of the input graph G using the algorithms with $O(n \log^3 n)$ space, as prescribed in Proposition 23 (set ε as a constant).

We then compute an $O(\sqrt{\log n})$-approximation to the best $(1/3)$-balanced HC-tree of H using the algorithm in Proposition 20.

To analyze the approximation ratio, the resulting $(1/3)$-balanced HC-tree by the algorithm in Proposition 20 is an $O(\sqrt{\log n})$ approximation of the optimal HC tree of H. Furthermore, by Theorem 2, the cost of any $(1/3)$-balanced tree in H remains within an $O(1)$-factor of its cost in G. Hence, we have an $O(\sqrt{\log n})$ approximation for $\mathsf{OPT}(G)$.

Finally, in exponential time, we can brute-force find the exact minimum $(1/3)$-balanced cut on every subgraph of H induced by the HC tree. By Lemma 19, the HC-tree is a 9-approximation of the optimal cost on H, which provides an $O(1)$-approximation for $\mathsf{OPT}(G)$ by Theorem 2. ■

Remark 5 *The algorithms can be extended to dynamic streams by increasing the space by some* $\text{polylog}(n)$ *factors and using randomization – we simply use a dynamic streaming algorithm of Ahn et al. (2012) for finding a cut sparsifier instead.*

3. A Lower Bound for Algorithms with $o(n)$ Memory

In Result 1, we showed that there is a semi-streaming algorithm for the hierarchical clustering problem that asymptotically achieves the best approximation ratio possible for offline hierarchical clustering on any graph. The number of passes used by this algorithm is clearly optimal and its space is just within log-factors of its output size, the HC-tree, and is thus again near-optimal.

Nevertheless, one could consider a potentially more space-efficient algorithm (e.g. $o(n)$-memory) for a simpler variant of the problem where the goal is to simply measure the "clusterability" of the input graph, i.e., estimate the *value* (cost) of the optimal solution as opposed to returning the entire tree. In this section, we prove that this seemingly easier problem still does not admit a better solution even when allowing $\text{polylog}\,(n)$-passes over the input! Formally,

Theorem 6 *Any streaming algorithm that can estimate the value of optimal hierarchical clustering on every n-vertex graphs with approximation ratio $o(\frac{\log n}{\log\log n})$ and* $\text{polylog}\,(n)$*-passes requires* $\Omega(n/\text{polylog}\,(n))$ *space.*

To prove this theorem, we use a reduction from the following variant of the *noisy cycle counting (NOC)* problem of Assadi and N (2021).

Proposition 7 *For infinitely many choices of n, k such that $k < \sqrt{n}$, the following is true. Suppose* ALG *is a p-pass s-space algorithm that distinguishes the following two families of graphs:*

– *a vertex-disjoint collection of 2 cycles of length $n/8$ each and $\frac{3n}{4k}$ paths of length k each;*

– *a vertex-disjoint collection of $\frac{n}{8k}$ cycles of length $2k$ each and $\frac{3n}{4k}$ paths of length k each.*

Then, we have that,

$$s = \Omega(\frac{1}{p^5} \cdot (n/k)^{1-\gamma\cdot\frac{p}{k}}),$$

for some absolute constant $\gamma \in (0, 1)$.[5]

5. The extra k-paths in the above family are what one considers "noise"; they are seemingly necessary for the proof of Proposition 7 itself and thus we need to prove the reductions *despite* the existence of these extra paths not *because* of their existence.

The proof of Theorem 6 is by showing that the value of best HC-tree for the two different families of graphs in Proposition 7 differ considerably (for proper choice of parameter k). Due to space limit, we defer the proof to D.

4. A Lower Bound for Algorithms with $\text{polylog}\, n$ Memory

In this section, we prove another lower bound that shows that when the space of the algorithm is restricted to just $\text{polylog}(n)$ bits, even distinguishing between 'highly clusterable' inputs versus ones that are 'very far from being clusterable' is not possible. In particular, we show that,

Theorem 8 *Any streaming algorithm that uses* $\text{polylog}\,(n)$ *space cannot estimate the* value *of hierarchical clustering with an approximation ratio of* $n^{1/2-\delta}$ *for any constant* $\delta > 0$ *with constant probability strictly better than half.*

The proof of Theorem 8 is by establishing a novel streaming lower bound of its own independent interest: no $\text{polylog}\,(n)$-space streaming algorithm can distinguish between inputs consisting of a single *expander* on the entire set of vertices versus a collection of $k = n^{1/2-o(1)}$ vertex-disjoint expanders. It is easy then to prove that the objective value of hierarchical clustering differs by a factor of $n^{1/2-o(1)}$ between the two cases which concludes the proof. Thus, the main contribution of our work on this front is to establish the mentioned streaming lower bound, formalized as follows.

Theorem 9 *For any* $\delta \in (0, 1/2)$*, any streaming algorithm with* $o(n^{\delta}/\log n)$ *space cannot distinguish these two families of* n*-vertex (multi-)graphs*[6] *with constant probability better than half:*

- ***Case 1:*** *A single expander* G *on* n *vertices and* $m = 10\, n \log n$ *edges;*
- ***Case 2:*** *A collection of* $t := n^{1/2-\delta}$ *vertex-disjoint expanders* G_i *each on* $n_i := n/t = n^{1/2+\delta}$ *vertices and* $m_i := 10\, n \log n/t = 10\, n^{1/2+\delta} \cdot \log n$ *edges.*

Here, by an expander, we mean a (multi-)graph with edge expansion *of* $\Omega(\log n)$ *as in Definition 21.*

The problem in Theorem 9 is qualitatively similar to the *gap cycle counting* problem studied extensively in the streaming literature (see, e.g., (Verbin and Yu, 2011; Kapralov et al., 2014; Assadi et al., 2020; Assadi and N, 2021; Kapralov et al., 2022)) wherein the goal is to distinguish between a single Hamiltonian cycle (or a few 'long' cycles) and a collection of vertex-disjoint 'short' cycles. Owing to its wide range of applications, the gap cycle counting problem has become a staple in graph streaming lower bounds. We believe our lower bound for the 'expander-variant' of this problem appears flexible enough to find other applications and is therefore interesting in its own right.

In the following, we first show how Theorem 8 follows easily from Theorem 9 and then concentrate the bulk of our effort in this section to proving the latter theorem.

Proof [Proof of Theorem 8 (assuming Theorem 9)] Suppose we have a streaming algorithm $\mathcal{A}$ that can estimate the value of hierarchical clustering for every graph G to within a factor $o(n^{1/2-\delta})$ with probability strictly more than half.

6. For technical reasons, we allow multi-graphs with edge multiplicity $O(1)$, which is standard; see, e.g. (Kapralov et al., 2014).

First, consider a graph G according to Case 1 of Theorem 9. We argue that in this case, $\mathsf{OPT}(G) = \Omega(n^2 \cdot \log n)$. By Lemma 19, we know that the algorithm that picks the minimum $1/3$-balanced cut repeatedly achieves an $O(1)$-approximation to $\mathsf{OPT}(G)$. At the same time, since edge expansion of G is $\Omega(\log n)$, for any $1/3$-balanced cut S, we have that $|\delta(S)| = \Omega(n \log n)$. Thus, the cost of that algorithm on its first level is already $\Omega(n^2 \log n)$, which gives $\mathsf{OPT}(G) = \Omega(n^2 \cdot \log n)$.

Now, consider a graph G according to Case 2 of Theorem 9. We have,

$$\mathsf{OPT}(G) = \sum_{i=1}^{t} \mathsf{OPT}(G_i) \leq t \cdot (n/t) \cdot (10n \log n/t) = O(n^2 \cdot \log n/t),$$

where the first equality is by Lemma 13 as G_i's are vertex-disjoint components of G, and the inequality is by Lemma 16 as each G_i contains (n/t) vertices and $(10n \log n/t)$ edges.

Combining the above two arguments, we have that $\mathsf{OPT}(G)$ differs by an $\Omega(t) = \Omega(n^{1/2-\delta})$ factor between the two cases of the problem in Theorem 9. Thus, $\mathcal{A}$ should be able to distinguish between these two cases with probability strictly more than half. By Theorem 9, we get that $\mathcal{A}$ has to have space $\Omega(n^{\delta}/\log n) \gg \mathrm{polylog}(n)$, concluding the proof. ■

4.1. A High-Level Overview of Proof of Theorem 9

The proof of Theorem 9 is via *communication complexity*, and then using the standard fact that communication complexity can lower bound the space of streaming algorithms. The communication complexity lower bound itself goes through several steps as we elaborate below.

Step one: a k-party communication problem. For integers $n, k, t \geq 1$, we define a k-party communication problem *One-vs-Many-Expander* ($\mathsf{OvME}_{n,k,t}$) on n-vertex graphs G. In $\mathsf{OvME}_{n,k,t}$, we have k players and each player P_i receives a *matching* M_i of size $n/4$ on n vertices. In addition, there exists a labeling Σ of vertices of G into t equal-size classes $\Sigma_1, \ldots, \Sigma_t$. Then,

- In the **Yes** case, the input matching of each player is chosen randomly, independent of Σ.
- In the **No** case, the input matching of each player is chosen randomly so that it contains $n/4t$ random edges inside each class Σ_j for $j \in [t]$.

The goal is for the players starting from P_1 to each send a message to the next player, so that the last player P_k can output which case the input belongs to.

We show that proving an $o(n^{\delta}/\log n)$ communication lower bound for $\mathsf{OvME}_{n,k,t}$ for $k = 40 \log n$ and $t = n^{1/2-\delta}$ implies Theorem 9. The proof is by showing that, with high probability, the **Yes**-case of OvME results in G corresponding to Case 1 of Theorem 9, while the **No**-case is the Case 2 of that theorem. This argument itself is a simple exercise in random graph theory.

Step two: a 2-party communication problem. In order to prove the lower bound for $\mathsf{OvME}_{n,k,t}$, we use a common approach (see, e.g., (Kapralov et al., 2014)) and reduce it to a 2-party problem which we call the *Hidden Labeling Problem* ($\mathsf{HLP}_{n,t}$) on n-vertex graphs G. In $\mathsf{HLP}_{n,t}$, Alice is given a labeling Σ of vertices of G into t equal-size classes $\Sigma_1, \ldots, \Sigma_t$ and Bob is given a single matching M of size $n/4$. The distribution of these labeling Σ and matching M is the same as the ones in $\mathsf{OvME}_{n,k,t}$ (where M can correspond to the input of any one player).

We prove that an $n^{\delta-o(1)}$ communication lower bound for $\mathsf{HLP}_{n,t}$ for protocols with probability of success $1/2+\Omega(1/k)$ implies our desired lower bound in the previous part for $\mathsf{OvME}_{n,k,t}$. The proof is via a hybrid argument over the input of k players in $\mathsf{OvME}_{n,k,t}$ similar to (Kapralov et al., 2014).

Step three: a decorrelation step. We note that the $\mathsf{HLP}_{n,t}$ problem is qualitatively similar to the famous *Boolean Hidden Matching* problem of (Gavinsky et al., 2007) and many of its variants such as *Boolean Hidden Partition* (Kapralov et al., 2014), or *p-ary Hidden Matching* (Guruswami and Tao, 2019), and alike (see, e.g., (Guruswami et al., 2017; Chou et al., 2020)). However, quantitatively, this problem is quite different from all these problems. For instance, all aforementioned problems admit an $\Omega(\sqrt{n})$ communication lower bound, while there is a protocol for solving HLP using $O(\sqrt{n/t}\cdot\log t)$ communication by focusing only on one class Σ_i in Alice's input[7]. As a result, while our lower proof for HLP borrows ideas from this line of work, and in particular the Fourier-analytic method of (Gavinsky et al., 2007), it also requires its own different ideas.

To prove the lower bound, we first 'break the (strong) correlation' on the edges of M_i in the input distribution (see, e.g., (Assadi and N, 2021) for a similar argument). This gives us yet another reduction to the following problem, which we denote by $\mathsf{HLP}^\star_m$: Alice is given an equipartition U_0, U_1 of m vertices and Bob is given a single edge e: In **Yes**-case, the edge e is chosen uniformly among all edges possible on $U_0 \cup U_1$, while in the **No**-case, the edge e is chosen uniformly from either edges entirely in U_0 or entirely in U_1. We show that for some $m=\Theta(n/t)$, any communication lower bound for $\mathsf{HLP}^\star_m$ for protocols with (quite low but non-trivial) probability of success of $1/2+\tilde{\Theta}(1/n)$, also implies the same lower bound for protocols for $\mathsf{HLP}_{n,t}$ that succeed with probability $1/2+\tilde{\Theta}(1)$. We shall note that technically speaking, here, we will not consider protocols that solve $\mathsf{HLP}^\star_m$ with certain probability, but rather the ones wherein *KL-divergence* of final 'view' of Bob in Yes- and No-cases differ by at least $\tilde{\Theta}(1/n)$. This will be crucial for the proof of our next step.

Step four: a low-probability-of-success lower bound. The very final step of our approach is to prove a lower bound for $\mathsf{HLP}^\star_m$ that rules out protocols where Bob's view is slightly different between Yes- and No-cases, namely, by $\tilde{\Theta}(1/n)$ in KL-divergence. This is done using a Fourier-analytic approach initiated in (Gavinsky et al., 2007), using the celebrated *KKL inequality* of (Kahn et al., 1988), that allows us to argue any protocol with c bits of communication for $\mathsf{HLP}^\star_m$ can only lead to an advantage of $O((c/m)^2)$ in changing Bob's view of which case the input belongs based on Alice's message.

Tracing back these parameters implies that to get an advantage of $\tilde{\Theta}(1/n)$ in solving $\mathsf{HLP}^\star_m$ (as dictated by step three), we need c to be:

$$c=\tilde{\Omega}(\frac{m}{\sqrt{n}})=\tilde{\Omega}(\frac{n}{t\sqrt{n}})=\tilde{\Omega}(\frac{\sqrt{n}}{t})=\tilde{\Omega}(n^{\delta}),$$

by the choice of $t=n^{1/2-\delta}$ in step one. By plugging in these bounds in the steps two and three, we get a lower bound of $\widetilde{\Omega}(n^{\delta})$ communication for $\mathsf{OvME}_{n,k,t}$ for any $k=\tilde{\Theta}(1)$ and $t=n^{1/2-\delta}$. Finally, such a lower bound by step one implies our desired streaming lower bound in Theorem 9. This concludes the high level overview of the proof of Theorem 9.

7. Alice sends $O(\sqrt{n/t})$ vertices of her input that belong to the class Σ_1; in the **Yes**-case, Bob is unlikely to have any edges inside this set, while in the **No**-case, one of Bob's edges will belong to this set with a high constant probability.

5. A Lower Bound for Exact Hierarchical Clustering Solution

One might be interested in using more memory to circumvent *any* approximation factor. In particular, a natural question to ask is if we can obtain the *exact* HC solution if we increase the memory to some value $o(n^2)$, which would still be non-trivial for space complexity. We answer the above question in the negative in this section in the following theorem.

Theorem 10 *Any single-pass streaming algorithm that outputs the optimal* value *of hierarchical clustering with probability at least* $2/3$ *requires a memory of* $\Omega(n^2)$ *bits even with unbounded computation time.*

Our lower bound effectively rules out any streaming algorithm that (asymptotically) outperforms the naive algorithm that stores every edge and solves the problem offline in exponential time. This further justifies the 'fitness' of our semi-streaming algorithm in Section 2.

High-level overview of the proof for Theorem 10. The lower bound follows from a reduction from the following variant of the well-known Index communication problem: let Alice's input be a random bipartite graph $G = (V, E)$ with each edge appearing with probability half, and let Bob's input be a vertex pair (i, j). Alice sends a message to Bob, and Bob is required to output whether $(i, j) \in E$. This problem is equivalent to the Index problem on a universe of size $\binom{n}{2}$ and thus requires $\Omega(n^2)$ communication (Ablayev, 1993).

We then reduce the problem to hierarchical clustering, which is the main technical step in the proof of Theorem 10. We provide a new construction that reduces the existence of edge (i, j) to the exact optimal HC cost by adding edges on Bob's side. In particular, for all vertices *except* i on the left partition, Bob connects them with a large clique; similarly, for all vertices *except* j in the right partition, Bob connects them with another large clique. Finally, Bob connects i and j respectively with a large clique, and he ensures the sizes of the four cliques are equal (see Figure 2). Ideally, if we can *control* the split pattern of the graphs constructed by the two players in the optimal HC tree, we can get that the optimal cost differ slightly based on the existence of edge (i, j). As such, Bob can use the exact optimal cost as a signal to distinguish the corresponding Index problem.

What remains is to understand the pattern of splits for a graph prescribed as above. To this end, we show a structural lemma that characterizes optimal HC trees on such graphs: we prove that the optimal tree always *first separates* the desired vertices pair into different components, and then split the rest of the graph in a *fixed order*. This is a generalization of the previous work of (Dasgupta, 2016) on computing the optimal HC trees of simpler graphs such as cliques and cycles.

En route to the proof of our main lower bound, we establish a weaker structural result that controls the pattern of split among *two* sparsely-connected cliques. This weaker result is necessary for the main proof of Theorem 10. We also note that the 'two-clique' version of the structural result already gives us a (weaker) $\Omega(n^2)$ lower bound for streaming algorithms that output the hierarchical clustering tree *with* the split costs.

Acknowledgments

We thank Sanjeev Khanna for communicating their results in Agarwal et al. (2022) to us and helpful conversations about their work and its connection to ours. Sepehr Assadi and Chen Wang are supported in part by a NSF CAREER Grant CCF-2047061, a gift from Google Research, and a Fulcrum award from Rutgers Research Council.

References

Amir Abboud, Vincent Cohen-Addad, and Hussein Houdrougé. Subquadratic high-dimensional hierarchical clustering. In *Advances in Neural Information Processing Systems*, pages 11576–11586, 2019.

Farid M. Ablayev. Lower bounds for one-way probabilistic communication complexity. In *Automata, Languages and Programming, 20nd International Colloquium, ICALP93, Lund, Sweden, July 5-9, 1993, Proceedings*, pages 241–252, 1993.

Arpit Agarwal, Sanjeev Khanna, Huan Li, and Prathamesh Patil. Sublinear algorithms for hierarchical clustering. *Manuscript*, 2022.

Kook Jin Ahn and Sudipto Guha. Graph sparsification in the semi-streaming model. In *Automata, Languages and Programming, 36th Internatilonal Colloquium, ICALP 2009, Rhodes, Greece, July 5-12, 2009, Proceedings, Part II*, pages 328–338, 2009.

Kook Jin Ahn, Sudipto Guha, and Andrew McGregor. Graph sketches: sparsification, spanners, and subgraphs. In *Proceedings of the 31st ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, PODS 2012, Scottsdale, AZ, USA, May 20-24, 2012*, pages 5–14, 2012. doi: 10.1145/2213556.2213560. URL http://doi.acm.org/10.1145/2213556.2213560.

Noga Alon, Yossi Matias, and Mario Szegedy. The space complexity of approximating the frequency moments. In *STOC*, pages 20–29. ACM, 1996.

Noga Alon, Yossi Azar, and Danny Vainstein. Hierarchical clustering: A 0.585 revenue approximation. In *Conference on Learning Theory*, pages 153–162. PMLR, 2020.

Sanjeev Arora, Elad Hazan, and Satyen Kale. 0(sqrt (log n)) approximation to SPARSEST CUT in õ(n^2) time. In *45th Symposium on Foundations of Computer Science (FOCS 2004), 17-19 October 2004, Rome, Italy, Proceedings*, pages 238–247. IEEE Computer Society, 2004. doi: 10.1109/FOCS.2004.1. URL https://doi.org/10.1109/FOCS.2004.1.

Sanjeev Arora, Satish Rao, and Umesh Vazirani. Expander flows, geometric embeddings and graph partitioning. *Journal of the ACM (JACM)*, 56(2):1–37, 2009.

Sepehr Assadi and Vishvajeet N. Graph streaming lower bounds for parameter estimation and property testing via a streaming XOR lemma. In Samir Khuller and Virginia Vassilevska Williams, editors, *STOC '21: 53rd Annual ACM SIGACT Symposium on Theory of Computing, Virtual Event, Italy, June 21-25, 2021*, pages 612–625. ACM, 2021. doi: 10.1145/3406325.3451110. URL https://doi.org/10.1145/3406325.3451110.

Sepehr Assadi, MohammadHossein Bateni, Aaron Bernstein, Vahab S. Mirrokni, and Cliff Stein. Coresets meet EDCS: algorithms for matching and vertex cover on massive graphs. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, pages 1616–1635, 2019a.

Sepehr Assadi, Yu Chen, and Sanjeev Khanna. Sublinear algorithms for $(\Delta + 1)$ vertex coloring. In Timothy M. Chan, editor, *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, pages 767–786. SIAM, 2019b.

Sepehr Assadi, Gillat Kol, Raghuvansh R. Saxena, and Huacheng Yu. Multi-pass graph streaming lower bounds for cycle counting, max-cut, matching size, and other problems. In Sandy Irani, editor, *61st IEEE Annual Symposium on Foundations of Computer Science, FOCS 2020, Durham, NC, USA, November 16-19, 2020*, pages 354–364. IEEE, 2020.

Dmitrii Avdiukhin, Sergey Pupyrev, and Grigory Yaroslavtsev. Multi-dimensional balanced graph partitioning via projected gradient descent. *Proc. VLDB Endow.*, 12(8):906–919, 2019.

Mohammadhossein Bateni, Soheil Behnezhad, Mahsa Derakhshan, MohammadTaghi Hajiaghayi, Raimondas Kiveris, Silvio Lattanzi, and Vahab Mirrokni. Affinity clustering: Hierarchical clustering at scale. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017a. URL https://proceedings.neurips.cc/paper/2017/file/2e1b24a664f5e9c18f407b2f9c73e821-Paper.pdf.

MohammadHossein Bateni, Soheil Behnezhad, Mahsa Derakhshan, MohammadTaghi Hajiaghayi, Raimondas Kiveris, Silvio Lattanzi, and Vahab Mirrokni. Affinity clustering: Hierarchical clustering at scale. In *Advances in Neural Information Processing Systems*, pages 6864–6874, 2017b.

Paul Beame, Paraschos Koutris, and Dan Suciu. Communication steps for parallel query processing. In *Proceedings of the 32nd ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, PODS 2013, New York, NY, USA - June 22 - 27, 2013*, pages 273–284, 2013.

András A. Benczúr and David R. Karger. Approximating *s-t* minimum cuts in $\tilde{O}(n^2)$ time. In Gary L. Miller, editor, *Proceedings of the Twenty-Eighth Annual ACM Symposium on the Theory of Computing, Philadelphia, Pennsylvania, USA, May 22-24, 1996*, pages 47–55. ACM, 1996. doi: 10.1145/237814.237827. URL https://doi.org/10.1145/237814.237827.

András A. Benczúr and David R. Karger. Randomized approximation schemes for cuts and flows in capacitated graphs. *SIAM J. Comput.*, 44(2):290–319, 2015.

Pavel Berkhin. A survey of clustering data mining techniques. In *Grouping multidimensional data*, pages 25–71. Springer, 2006.

Luigi L Cavalli-Sforza and Anthony WF Edwards. Phylogenetic analysis: models and estimation procedures. *Evolution*, 21(3):550–570, 1967.

Ines Chami, Albert Gu, Vaggos Chatziafratis, and Christopher Ré. From trees to continuous embeddings and back: Hyperbolic hierarchical clustering. *Advances in Neural Information Processing Systems*, 33:15065–15076, 2020.

Moses Charikar and Vaggos Chatziafratis. Approximate hierarchical clustering via sparsest cut and spreading metrics. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 841–854. SIAM, 2017.

Moses Charikar, Vaggos Chatziafratis, and Rad Niazadeh. Hierarchical clustering better than average-linkage. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2291–2304. SIAM, 2019.

Vaggos Chatziafratis, Rad Niazadeh, and Moses Charikar. Hierarchical clustering with structural constraints. In *International Conference on Machine Learning*, pages 774–783, 2018.

Vaggos Chatziafratis, Mohammad Mahdian, and Sara Ahmadian. Maximizing agreements for ranking, clustering and hierarchical clustering via max-cut. In *International Conference on Artificial Intelligence and Statistics*, pages 1657–1665. PMLR, 2021.

Chi-Ning Chou, Alexander Golovnev, and Santhoshini Velusamy. Optimal streaming approximations for all boolean max-2csps and max-ksat. In Sandy Irani, editor, *61st IEEE Annual Symposium on Foundations of Computer Science, FOCS 2020, Durham, NC, USA, November 16-19, 2020*, pages 330–341. IEEE, 2020. doi: 10.1109/FOCS46700.2020.00039. URL https://doi.org/10.1109/FOCS46700.2020.00039.

Chi-Ning Chou, Alexander Golovnev, Madhu Sudan, and Santhoshini Velusamy. Approximability of all finite csps in the dynamic streaming setting. *Electron. Colloquium Comput. Complex.*, page 63, 2021.

Vincent Cohen-Addad, Varun Kanade, Frederik Mallmann-Trenn, and Claire Mathieu. Hierarchical clustering: Objective functions and algorithms. *Journal of the ACM (JACM)*, 66(4):1–42, 2019.

Thomas M. Cover and Joy A. Thomas. *Elements of information theory (2. ed.)*. Wiley, 2006. ISBN 978-0-471-24195-9.

Artur Czumaj, Jakub Lacki, Aleksander Madry, Slobodan Mitrovic, Krzysztof Onak, and Piotr Sankowski. Round compression for parallel matching algorithms. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, June 25-29, 2018*, pages 471–484, 2018.

Sanjoy Dasgupta. A cost function for similarity-based hierarchical clustering. In Daniel Wichs and Yishay Mansour, editors, *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*, pages 118–127. ACM, 2016. doi: 10.1145/2897518.2897527. URL https://doi.org/10.1145/2897518.2897527.

Laxman Dhulipala, David Eisenstat, Jakub Łącki, Vahab Mirrokni, and Jessica Shi. Hierarchical agglomerative graph clustering in nearly-linear time. In *International Conference on Machine Learning*, pages 2676–2686. PMLR, 2021.

Michael B Eisen, Paul T Spellman, Patrick O Brown, and David Botstein. Cluster analysis and display of genome-wide expression patterns. *Proceedings of the National Academy of Sciences*, 95(25):14863–14868, 1998.

Joan Feigenbaum, Sampath Kannan, Andrew McGregor, Siddharth Suri, and Jian Zhang. On graph problems in a semi-streaming model. *Theor. Comput. Sci.*, 348(2-3):207–216, 2005. doi: 10.1016/j.tcs.2005.09.013. URL http://dx.doi.org/10.1016/j.tcs.2005.09.013.

Joseph Felsenstein. *Inferring phylogenies*, volume 2. Sinauer associates Sunderland, MA, 2004.

Dmitry Gavinsky, Julia Kempe, Iordanis Kerenidis, Ran Raz, and Ronald de Wolf. Exponential separations for one-way quantum communication complexity, with applications to cryptography. In David S. Johnson and Uriel Feige, editors, *Proceedings of the 39th Annual ACM Symposium on Theory of Computing, San Diego, California, USA, June 11-13, 2007*, pages 516–525. ACM, 2007. doi: 10.1145/1250790.1250866. URL https://doi.org/10.1145/1250790.1250866.

Venkatesan Guruswami and Runzhou Tao. Streaming hardness of unique games. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2019, September 20-22, 2019, Massachusetts Institute of Technology, Cambridge, MA, USA*, pages 5:1–5:12, 2019.

Venkatesan Guruswami, Ameya Velingker, and Santhoshini Velusamy. Streaming complexity of approximating max 2csp and max acyclic subgraph. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2017, August 16-18, 2017, Berkeley, CA, USA*, pages 8:1–8:19, 2017.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer, 2nd edition, 2009.

Jeff Kahn, Gil Kalai, and Nathan Linial. The influence of variables on boolean functions (extended abstract). In *29th Annual Symposium on Foundations of Computer Science, White Plains, New York, USA, 24-26 October 1988*, pages 68–80. IEEE Computer Society, 1988. doi: 10.1109/SFCS.1988.21923. URL https://doi.org/10.1109/SFCS.1988.21923.

Michael Kapralov and Dmitry Krachun. An optimal space lower bound for approximating MAX-CUT. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, STOC 2019, Phoenix, AZ, USA, June 23-26, 2019*, pages 277–288, 2019.

Michael Kapralov, Sanjeev Khanna, and Madhu Sudan. Approximating matching size from random streams. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014, Portland, Oregon, USA, January 5-7, 2014*, pages 734–751, 2014.

Michael Kapralov, Sanjeev Khanna, and Madhu Sudan. Streaming lower bounds for approximating MAX-CUT. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2015, San Diego, CA, USA, January 4-6, 2015*, pages 1263–1282, 2015.

Michael Kapralov, Amulya Musipatla, Jakab Tardos, David P. Woodruff, and Samson Zhou. Noisy boolean hidden matching with applications. In Mark Braverman, editor, *13th Innovations in Theoretical Computer Science Conference, ITCS 2022, January 31 - February 3, 2022, Berkeley, CA, USA*, volume 215 of *LIPIcs*, pages 91:1–91:19. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022. doi: 10.4230/LIPIcs.ITCS.2022.91. URL https://doi.org/10.4230/LIPIcs.ITCS.2022.91.

Howard J. Karloff, Siddharth Suri, and Sergei Vassilvitskii. A model of computation for mapreduce. In *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2010, Austin, Texas, USA, January 17-19, 2010*, pages 938–948, 2010.

Matthäus Kleindessner and Ulrike von Luxburg. Kernel functions based on triplet comparisons. In *Advances in Neural Information Processing Systems*, pages 6810–6820, 2017.

Ravi Kumar, Benjamin Moseley, Sergei Vassilvitskii, and Andrea Vattani. Fast greedy algorithms in mapreduce and streaming. In *25th ACM Symposium on Parallelism in Algorithms and Architectures, SPAA '13, Montreal, QC, Canada - July 23 - 25, 2013*, pages 1–10, 2013.

Yaniv Loewenstein, Elon Portugaly, Menachem Fromer, and Michal Linial. Efficient algorithms for accurate hierarchical clustering of huge datasets: tackling the entire protein space. *Bioinformatics*, 24(13):i41–i49, 2008.

Charles F Mann, David W Matula, and Eli V Olinick. The use of sparsest cuts to reveal the hierarchical community structure of social networks. *Social Networks*, 30(3):223–234, 2008.

Andrew McGregor. Graph stream algorithms: a survey. *SIGMOD Record*, 43(1):9–20, 2014. URL http://doi.acm.org/10.1145/2627692.2627694.

Nicholas Monath, Ari Kobren, Akshay Krishnamurthy, Michael R Glass, and Andrew McCallum. Scalable hierarchical clustering with tree grafting. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1438–1448, 2019a.

Nicholas Monath, Manzil Zaheer, Daniel Silva, Andrew McCallum, and Amr Ahmed. Gradient-based hierarchical clustering using continuous representations of trees in hyperbolic space. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 714–722, 2019b.

Nicholas Monath, Kumar Avinava Dubey, Guru Guruganesh, Manzil Zaheer, Amr Ahmed, Andrew McCallum, Gokhan Mergen, Marc Najork, Mert Terzihan, Bryon Tjanaka, et al. Scalable hierarchical agglomerative clustering. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 1245–1255, 2021.

Benjamin Moseley and Joshua Wang. Approximation bounds for hierarchical clustering: Average linkage, bisecting k-means, and local search. In *Advances in Neural Information Processing Systems*, pages 3094–3103, 2017.

S Muthukrishnan. *Data streams: Algorithms and applications*. Now Publishers Inc, 2005.

Stanislav Naumov, Grigory Yaroslavtsev, and Dmitrii Avdiukhin. Objective-based hierarchical clustering of deep embedding vectors. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 9055–9063, 2021.

Maximillian Nickel and Douwe Kiela. Poincaré embeddings for learning hierarchical representations. In *Advances in Neural Information Processing Systems*, pages 6338–6347, 2017.

Anand Rajagopalan, Fabio Vitale, Danny Vainstein, Gui Citovsky, Cecilia M Procopiuc, and Claudio Gentile. Hierarchical clustering of data streams: Scalable algorithms and approximation guarantees. In *International Conference on Machine Learning*, pages 8799–8809. PMLR, 2021.

Aurko Roy and Sebastian Pokutta. Hierarchical clustering via spreading metrics. In *Advances in Neural Information Processing Systems*, pages 2316–2324, 2016.

Baris Sumengen, Anand Rajagopalan, Gui Citovsky, David Simcha, Olivier Bachem, Pradipta Mitra, Sam Blasiak, Mason Liang, and Sanjiv Kumar. Scaling hierarchical agglomerative clustering to billion-sized datasets. *arXiv preprint arXiv:2105.11653*, 2021.

Michele Tumminello, Fabrizio Lillo, and Rosario N Mantegna. Correlation, hierarchies, and networks in financial markets. *Journal of economic behavior & organization*, 75(1):40–58, 2010.

Elad Verbin and Wei Yu. The streaming complexity of cycle counting, sorting by reversals, and other problems. In *Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2011, January 23-25, 2011*, pages 11–25, 2011.

Sharad Vikram and Sanjoy Dasgupta. Interactive Bayesian hierarchical clustering. In *International Conference on Machine Learning*, pages 2081–2090, 2016.

Kiri Wagstaff and Claire Cardie. Clustering with instance-level constraints. *AAAI/IAAI*, 1097:577–584, 2000.

Kiri Wagstaff, Claire Cardie, Seth Rogers, and Stefan Schrödl. Constrained k-means clustering with background knowledge. In *ICML*, volume 1, pages 577–584, 2001.

Appendix A. Preliminaries and Standard Results for Hierarchical Clustering

Notation. As standard in the literature, we denote a graph $G = (V, E, w)$ with V as the set of the vertices, E as the set of the edges, and $w : E \to \mathbb{R}^+$ be the edge weights. For any subset of vertices $A \subseteq V$, we use $\bar{A} = V \setminus A$ to denote the complementary set of vertices in G. We refer to any disjoint sets $A, B \subseteq V$ of vertices in V as a *cut* (A, B). If we further have $B = \bar{A}$ (i.e., cut $(A, V \setminus A)$), then the cut is called a *global* cut. For a cut (A, B), the set of *cut edges* is the set of edges that are between A and B, denoted by $\delta(A, B)$. We denote the weight of a cut as $w(A, B) = \sum_{e\in\delta(A,B)} w(e)$.

A.1. Standard Results on Dasgupta's Hierarchical Clustering Cost

There have been a fruitful collection of results on understanding the cost function in Eq (1) since the work of Dasgupta Dasgupta (2016). In this section, we present some known results for hierarchical clustering that lay the foundations of our paper.

OPTIMAL HIERARCHICAL CLUSTERING TREES

We first give a collection of lemmas that characterize the behavior for optimal HC trees. The proofs of the lemmas can be found in Appendix A.

We start with the following observations for the optimal costs of the HC trees.

Observation 11 (Dasgupta (2016)) *Suppose G is any graph, A and $\bar{A}$ are two disjoint subsets of vertices in G, and G_A and $G_{\bar{A}}$ are induced subgraphs of G on vertices A and $\bar{A}$, respectively. Then,*

$$\mathsf{OPT}(G_A) + \mathsf{OPT}(G_{\bar{A}}) \leq \mathsf{OPT}(G).$$

Observation 12 (Dasgupta (2016)) *Let G be any graph and $\mathcal{T}$ be a HC tree for G. Then, for every $\mathcal{T}$ that is not binary, there exists a binary $\mathcal{T}'$ such that* $\mathbf{cost}_G(\mathcal{T}') \leq \mathbf{cost}_G(\mathcal{T})$.

Observations 11 and 12 are among the first results of Dasgupta's hierarchical clustering, and the proofs can be found in Dasgupta (2016). In Observation 11, the 'equals to' relation is attained by graphs of vertex-disjoint disconnected components. More formally, we have

Lemma 13 (Dasgupta (2016)) *Let G be a vertex-disjoint union of graphs A, B. Then,*

$$\mathsf{OPT}(G) = \mathsf{OPT}(A) + \mathsf{OPT}(B).$$

Proof By Observation 12, an optimal hierarchical clustering tree $\mathcal{T}^*$ never partition the graph into more than two disconnected components. Therefore, if G is a vertex-disjoint union of graphs, which means A and B are disconnected, the optimal tree always first separates A and B. Furthermore, since we have $w(A, B) = 0$, this separation induces a 0 cost, which implies the lemma statement. ■

As a result of Lemma 13, for any optimal HC tree on vertex-disjoint union of graphs A, B, the top-level node always splits A and B.

The following lemmas capture the optimal HC costs on paths and cycles.

Lemma 14 (Dasgupta (2016)) *Let P_m denote a path of length m. Then, $\mathsf{OPT}(P_m) = m \log m + O(m)$.*

The basic idea for Lemma 14 is that since an optimal tree is always binary (Observation 12), the optimal strategy is to 'balance' the cost at each level and the cost it incurred for all lower levels. Therefore, by a balanced-tree recursion argument, the optimal cost for splitting a line is to *always split as balanced as possible*, which results in a cost of $m \log m + O(m)$.

Lemma 15 (Dasgupta (2016)) *Let C_m denote a cycle of length m. Then, we have*

$$\mathsf{OPT}(C_m) = m \log(m) + O(m).$$

Proof We can reduce the cost of splitting a cycle of length m to the case of a line. Note that the optimal tree is always binary. Therefore, in the first split, the optimal Hierarchical Clustering always splits the cycle into two disjoint parts (either 2 paths or an length $(m-1)$ cycle and a singleton vertex). Therefore, one can write the objective of the cost as

$$\textbf{cost}(C_m) = 2m + \min_{\substack{0 \le m_1 \le m-2 \\ 0 \le m_2 \le m-2 \\ m_1 + m_2 = m-2}} \mathsf{OPT}(P_{m_1}) + \mathsf{OPT}(P_{m_2}),$$

where the optimum is attained when $m_1 = m_2 = \frac{m}{2} - 1$. Plugging in the optimal values of m_1 and m_2 leads to the desired statement. ■

Finally, we have the following trivial upper bound on the maximum costs of HC on any graph, by simply splitting all edges in the first level.

Fact 16 *For any graph G with m edges and n vertices, $\mathsf{OPT}(G) \le m \cdot n$.*

HIERARCHICAL CLUSTERING COST AS A FUNCTION OF CUTS

We now show that the cost function in Eq (1) can be represented as a function of cuts in the subgraphs of G. By Observation 12, we can assume w.log. that the HC-tree is binary. For each non-leaf-node z of $\mathcal{T}$, we associate a cut (A, B), denoted by $\textbf{cut}(\mathcal{T}[z])$. Let z_1 and z_2 be the child nodes of z in $\mathcal{T}$, such that $|\textbf{leaf-nodes}(\mathcal{T}(z_1))| \leq |\textbf{leaf-nodes}(\mathcal{T}(z_2))|$. Then, we set $A := \textbf{leaf-nodes}(\mathcal{T}(z_1))$ and $B := \textbf{leaf-nodes}(\mathcal{T}(z_2))$. Observe that in Eq (1), the multiplicative factor of $w(e)$ for each edge $e = (u, v) \in E$ is equal to the minimum cluster size for u and v to be in the same cluster. Hence, we can alternatively write $\textbf{cost}_G(\mathcal{T})$ in Eq (1) as follows:

$$\textbf{cost}_G(\mathcal{T}) = \sum_{\substack{(A,B) := \textbf{cut}(\mathcal{T}[z]) \text{ for} \\ \text{internal nodes } z \text{ of } \mathcal{T}}} w(A, B) \cdot |A \cup B| \,. \tag{2}$$

APPROXIMATELY OPTIMAL HIERARCHICAL CLUSTERING TREES AS BALANCED TREES

Dasgupta's work proved that finding the optimal trees for the hierarchical clustering function is NP-hard (Dasgupta, 2016). Therefore, major efforts to study efficient HC algorithms have been devoted to *approximation algorithms*. It is known that we can find an approximation of the optimal hierarchical clustering by recursively applying approximate *balanced minimum cuts* on the graph. More formally, we define balanced cuts and balanced trees as follows.

Definition 17 (β-Balanced Cuts and Trees) *For any parameter β such that $0 < \beta < 1$, we say that a cut (A, B) is* β-balanced *if*

$$\max\{|A|, |B|\} \leq (1 - \beta) \cdot |A \cup B| \,.$$

A β-balanced cut (A, B) is said to be a β-balanced minimum cut if for any β-balanced cut $(A', B') \neq (A, B)$, there is $w(A, B) \leq w(A', B')$. Moreover, we say a HC tree $\mathcal{T}$ is a β-balanced tree *if for every non-leaf node z of $\mathcal{T}$,* $\textbf{cut}(\mathcal{T}[z])$ *is β-balanced.*

One way to create β-balanced trees is to recursively apply the β-balanced minimum cuts to the induced subgraphs, formally defined as follows.

Definition 18 (Recursive β-balanced Min-cut Procedure) *We say a HC tree $\mathcal{T}$ is obtained by the* recursive β-balanced min-cut procedure *on G if for each non-leaf node z of $\mathcal{T}$, the* $\textbf{cut}(\mathcal{T}[z])$ *is obtained by a β-balanced minimum cut (A, B) on the subgraph induced by $\mathcal{T}[z]$.*

It is known by Charikar and Chatziafratis (2017) that if one applies the procedure in Definition 18, it is possible to get a constant approximation of the optimal HC tree.

Lemma 19 (cf. Charikar and Chatziafratis (2017)) *For any graph $G = (V, E, w)$, let $\mathcal{T}_{balanced}$ be a $(1/3)$-balanced tree obtained by the procedure in Definition 18 with $\beta = \frac{1}{3}$. There is*

$$\textbf{cost}_G(\mathcal{T}_{balanced}) \leq 9 \cdot \mathsf{OPT}(G).$$

Lemma 19 was previously proved in Charikar and Chatziafratis (2017) with an unspecified constant ($O(1)$), and we provide a self-contained proof with the exact constant in Appendix C.2.

Note that Lemma 19 is structural and computing balanced minimum cut itself is not an easy task. Indeed, finding the exact balanced minimum cut is a NP-hard problem. However, it is known that one can compute an $O(\sqrt{\log n})$ approximation for balanced minimum cut in polynomial time by Arora et al. (2004). As such, we can obtain an $O(\sqrt{\log n})$-approximation algorithm for $\mathsf{OPT}(G)$ by recursively applying the $O(\sqrt{\log n})$-approximate $1/3$-balanced cut.

Proposition 20 (cf. Charikar and Chatziafratis (2017)) *There exists a polynomial-time algorithm that given a weighted undirected graph $G = (V, E, w)$, computes a $\frac{1}{3}$-balanced HC-tree $\mathcal{T}$ such that*

$$\mathbf{cost}_G(\mathcal{T}) \leq O(\sqrt{\log n}) \cdot \mathsf{OPT}(G).$$

We defer the proof of Proposition 20 to Appendix C.2.

A.2. Basic Graph Algorithms Backgrounds

In this section, we review a few standard graph algorithm definitions and results related to graph expansion and cut sparsifiers.

GRAPH EXPANSION

For a weighted graph $G = (V, E, w)$, we define the graph (edge) expansion as follows.

Definition 21 *The edge expansion of a graph $G = (V, E, w)$ is*

$$\Phi_G = \min_{\substack{S \subseteq V \\ |S| \leq \frac{n}{2}}} \frac{w(S, \bar{S})}{|S|},$$

where $w(S, \bar{S})$ is the total edge weights between S and $\bar{S}$.

The notion of edge expansion gives us a convenient tool to control the upper and lower bound of the hierarchical clustering cost, which is crucial to our proof in Section 4.

CUT SPARSIFIERS

We now describe the notion of cut sparsifers. On the high level, a cut sparsifier aims to 'sparsify' the edges by redistributing the weights to certain 'key edges'. By only storing a substantially smaller number of edges, the resulting graph can still maintain the weight of any *global* cut by a small approximation factor. Formally,

Definition 22 (Cut Sparsifier) *Given a graph $G = (V, E, w_G)$, we say that a weighted subgraph $H := (V, E_H, w_H)$ is a $(1 \pm \varepsilon)$-cut sparsifier of G if for all non-empty $A \subset V$, the following holds:*

$$(1 - \varepsilon) \cdot w_G(A, \bar{A}) \leq w_H(A, \bar{A}) \leq (1 + \varepsilon) \cdot w_G(A, \bar{A}),$$

where $w_G(A, \bar{A})$ (resp. $w_H(A, \bar{A})$) denotes the weight of cut-edges in $(A, \bar{A})$ in G (resp. in H).

The work by Benczúr and Karger (1996) first shows that such a spasifier exists for any graph, and it can be constructed in polynomial time. Furthermore, in the graph streaming model, it is known that with $\widetilde{O}(n)$ memory, one can achieve an ε-sparsifier in a *single* pass.

Proposition 23 (Ahn and Guha (2009); McGregor (2014)) *There exists a single-pass streaming algorithm that given a graph $G = (V, E, w)$, computes a $(1 \pm \varepsilon)$-cut spasifier of G with a memory of $O(\frac{n \log^3(n)}{\varepsilon^2})$ words.*

A.3. Basic Background on Information-Theory and Fourier Analysis

Finally, we review basic definitions from information-theory and Fourier analysis that we use in our paper. Appendix B.2 contains the details on the the key properties of these definition.

Definition 24 (KL-divergence) *Let X and Y be two discrete random variables supported over the domain Ω with distributions μ_X and μ_Y. The* **KL-divergence** *between X and Y is defined as*

$$\mathbb{D}(X \mid\mid Y) := \sum_{\omega \in \Omega} \mu_X(\omega) \log\left(\frac{\mu_X(\omega)}{\mu_Y(\omega)}\right).$$

We shall note that KL-divergence does *not* satisfy triangle inequality; however, it does admit a chain-rule which plays an important role in our proofs.

Definition 25 *Let X and Y be two discrete random variables supported over the domain Ω with distributions μ_X and μ_Y. The* **total variation distance (TVD)** *between X and Y is defined as*

$$\|X - Y\|_{\mathrm{tvd}} := \frac{1}{2} \sum_{\omega \in \Omega} |\mu_X(\omega) - \mu_Y(\omega)|.$$

The total variation distance is a metric and satisfies triangle inequality; it is also closely related to the probability of success of a *Maximum Likelihood Estimator* (MLE) for distinguishing a source of a sample. TVD can be upper bound via KL-divergence by Pinsker's inequality.

Finally, we use the following definition of Fourier transform on Boolean hypercube.

Definition 26 *The* **Fourier transform** *of a function $f : \{0,1\}^n \to \mathbb{R}$ is a function $\hat{f} : 2^{[n]} \to \mathbb{R}$:*

$$\hat{f}(S) := \sum_{x \in \{0,1\}^n} \frac{1}{2^n} \cdot f(x) \cdot \mathcal{X}_S(x),$$

where $\mathcal{X}_S(x) := (-1)^{\sum_{i \in S} x_i}$. We refer to each $\hat{f}(S)$ as a **Fourier coefficient***.*

Appendix B. Standard Technical Tools

B.1. Concentration Inequalities

We now present the standard concentration inequalities used in our proofs. We start from the following standard variant of Chernoff-Hoeffding bound.

Proposition 27 (Chernoff-Hoeffding bound) *Let $X_1, \ldots, X_n$ be n independent random variables with support in $[0, 1]$. Define $X := \sum_{i=1}^{n} X_i$. Then, for every $\delta \in (0, 1]$, there is*

$$\Pr\left(|X - \mathbb{E}[X]| > \delta \cdot \mathbb{E}[X]\right) \leq 2 \cdot \exp\left(-\frac{\delta^2\, \mathbb{E}[X]}{3}\right).$$

The standard Chernoff bound works on *independent* random variables. Going beyond the independent case, it is also known that Chernoff bound applies to *negatively correlated* random variables. Informally speaking, two random variables X_i and X_2 are negatively correlated if conditioning on $X_i = 1$, the probability for $X_j = 1$ decreases. Formally, we define negatively correlated random variables as follows.

Definition 28 (Negatively Correlated Random Variables) *Random variables $X_1, \ldots, X_n$ are said to be* negatively correlated *if and only if*

$$\mathbb{E}\left[\prod_{i=1}^{n} X_i\right] \leq \prod_{i=1}^{n} \mathbb{E}[X_i].$$

In particular, if X_i's are independent, we have $\mathbb{E}\left[\prod_{i=1}^{n} X_i\right] = \prod_{i=1}^{n} \mathbb{E}[X_i]$.

Proposition 29 (Generalized Chernoff) *Let $X_1, \ldots, X_n$ be n negatively correlated random variables supported on $\{0, 1\}$. Then, the concentration inequality in Proposition 27 still holds.*

B.2. Standard Tools for Lower Bound Proofs

We shall use the following standard properties of KL-divergence and TVD defined in Appendix A.3. For the proof of this results, see the excellent textbook by Cover and Thomas Cover and Thomas (2006).

The following facts state the chain rule property and convexity of KL-divergence.

Fact 30 (Chain rule of KL divergence) *For any random variables $X = (X_1, X_2)$ and $Y = (Y_1, Y_2)$ be two random variables,*

$$\mathbb{D}(X \,||\, Y) = \mathbb{D}(X_1 \,||\, Y_1) + \mathop{\mathbb{E}}_{x \sim X_1} \mathbb{D}(X_2 \mid X_1 = x \,||\, Y_2 \mid Y_1 = x).$$

Fact 31 (Convexity KL-divergence) *For any distributions μ_1, μ_2 and ν_1, ν_2 and any $\lambda \in (0, 1)$,*

$$\mathbb{D}(\lambda \cdot \mu_1 + (1 - \lambda) \cdot \mu_2 \,||\, \lambda \cdot \nu_1 + (1 - \lambda) \cdot \nu_2) \leq \lambda \cdot \mathbb{D}(\mu_1 \,||\, \nu_1) + (1 - \lambda) \cdot \mathbb{D}(\mu_2 \,||\, \nu_2).$$

Fact 32 (Conditioning cannot decrease KL-divergence) *For any random variables X, Y, Z,*

$$\mathbb{D}(X \,||\, Y) \leq \mathop{\mathbb{E}}_{z \sim Z} \mathbb{D}(X \mid Z = z \,||\, Y \mid Z = z).$$

Pinsker's inequality relates KL-divergence to TVD.

Fact 33 (Pinsker's inequality) *For any random variables X and Y supported over the same Ω,*

$$\|X - Y\|_{\text{tvd}} \leq \sqrt{\frac{1}{2} \cdot \mathbb{D}(X \mid\mid Y)}.$$

The following fact characterizes the error of MLE for the source of a sample based on the TVD of the originating distributions.

Fact 34 *Suppose μ and ν are two distributions over the same support Ω; then, given one sample s from either μ or ν, the best probability we can decide whether s came from μ or ν is*

$$\frac{1}{2} + \frac{1}{2} \cdot \|\mu - \nu\|_{\text{tvd}}.$$

Fourier analysis on Boolean hypercube. For any two functions $f, g : \{0,1\}^n \to \mathbb{R}$, we define the *inner product* between f and g as:

$$\langle f, g \rangle = \mathop{\mathbb{E}}_{x \in \{0,1\}^n} [f(x) \cdot g(x)] = \sum_{x \in \{0,1\}^n} \frac{1}{2^n} \cdot f(x) \cdot g(x).$$

For a set $S \subseteq \{0,1\}$, we define the *character* function $\mathcal{X}_S : \{0,1\}^n \to \{-1,+1\}$ as:

$$\mathcal{X}_S(x) = (-1)^{(\sum_{i \in S} x_i)} = \begin{cases} 1 & \text{if } \oplus_{i \in S}\, x_i = 0 \\ -1 & \text{if } \oplus_{i \in S}\, x_i = 1 \end{cases}.$$

The *Fourier transform* of $f : \{0,1\}^n \to \mathbb{R}$ is a function $\hat{f} : 2^{[n]} \to \mathbb{R}$ such that:

$$\hat{f}(S) = \langle f, \mathcal{X}_S \rangle = \sum_{x \in \{0,1\}^n} \frac{1}{2^n} \cdot f(x) \cdot \mathcal{X}_S(x).$$

We refer to each $\hat{f}(S)$ as a *Fourier coefficient*.

We use *KKL inequality* of Kahn et al. (1988) for bounding sum of *squared* of Fourier coefficients.

Proposition 35 (Kahn et al. (1988)) *For every function $f \in \{0,1\}^n \to \{-1,0,+1\}$ and every $\gamma \in (0,1)$*

$$\sum_{S \subseteq [n]} \gamma^{|S|} \cdot \hat{f}(S)^2 \leq \left(\frac{\text{supp}(f)}{2^n}\right)^{\frac{2}{1+\gamma}}.$$

Appendix C. Missing proofs related to the result in Section 2

C.1. Missing proof of Lemma 3

Proof Firstly, by Eq (2),

$$\mathbf{cost}_G(\mathcal{T}) = \sum_{\substack{(A,B) := \mathbf{cut}(\mathcal{T}[u]) \text{ for} \\ \text{internal nodes } u \text{ of } \mathcal{T}}} w_G(A,B) \cdot |A \cup B|$$

$$\leq \sum_{\substack{(A,B) := \textbf{cut}(\mathcal{T}[u]) \text{ for} \\ \text{internal nodes } u \text{ of } \mathcal{T}}} \frac{1}{2} \cdot (w_G(A, \bar{A}) + w_G(B, \bar{B})) \cdot |A \cup B| = W(G),$$

simply because $w_G(A, \bar{A}), w_G(B, \bar{B}) \geq w_G(A, B)$ as the set of edges in each term of the LHS is a superset of edges in RHS. This proves the first (and easy) part of the lemma.

We now show that the parameter $W(G)$ is also not much larger than $\textbf{cost}_G(\mathcal{T})$. Fix an edge $e = (u, v) \in E$. Let $P(u)$ and $P(v)$ denote the leaf-to-root paths of u and v in $\mathcal{T}$, respectively. Additionally, let

$$P^*(u) := \{w \in P(u) \mid w \neq u, v \notin \textbf{leaf-nodes}(\mathcal{T}[w])\},$$
$$P^*(v) := \{w \in P(v) \mid w \neq v, u \notin \textbf{leaf-nodes}(\mathcal{T}[w])\}.$$

That is, the paths $P^*(u)$ and $P^*(v)$ are the portions of $P(u)$ and $P(v)$, which are *strictly* between the leaves and $u \vee v$.

With these definitions, we can alternatively write $W(G)$ as:

$$W(G) = \sum_{e=(u,v)\in E} w(e) \cdot \left(|\textbf{leaf-nodes}(\mathcal{T}[u \vee v])| + \frac{1}{2} \sum_{w \in P^*(u) \cup P^*(v)} |\textbf{leaf-nodes}(\mathcal{T}[w])| \right).$$

This is because in each of the nodes $w \in P^*(u)$, u belongs to either A or B, while v does not belong to either, and thus we get a contribution of $\frac{1}{2}w(e)$ in exactly one of $w(A, \bar{A})$ or $w(B, \bar{B})$; this is similarly the case for nodes in $P^*(v)$; finally, $u \vee v$ is the only other node that splits u and v and in this case e contributes $\frac{1}{2}w(e)$ to both $w(A, \bar{A})$ and $w(B, \bar{B})$. See Figure 1 for an illustration.

We now use the balancedness of $\mathcal{T}$ to simplify the above bound further. Let $P^*(u) = (w_1, \ldots, w_k)$, with w_k being a child-node of $u \vee v$, which, for simplicity of notation, we denote by w_{k+1}. Considering $\mathcal{T}$ is β-balanced, we have that for every $i \in [k]$,

$$|\textbf{leaf-nodes}(\mathcal{T}[w_i])| \leq (1 - \beta) \cdot |\textbf{leaf-nodes}(\mathcal{T}[w_{i+1}])|.$$

As such, $|\textbf{leaf-nodes}(\mathcal{T}[w_i])|$ forms a geometric series and we thus have,

$$\begin{aligned}\sum_{i=1}^{k} |\textbf{leaf-nodes}(\mathcal{T}[w_i])| &\leq \sum_{i=1}^{\infty} (1-\beta)^i \, |\textbf{leaf-nodes}(\mathcal{T}[w_{k+1}])| \\ &= \frac{1-\beta}{\beta} \cdot |\textbf{leaf-nodes}(\mathcal{T}[w_{k+1}])| \\ &= \frac{1-\beta}{\beta} \cdot |\textbf{leaf-nodes}(\mathcal{T}[u \vee v])|.\end{aligned}$$

This can similarly be done for $P^*(v)$, thus leaving us with:

$$\frac{1}{2} \sum_{w \in P^*(u) \cup P^*(v)} |\textbf{leaf-nodes}(\mathcal{T}[w])| \leq \frac{1-\beta}{\beta} \cdot |\textbf{leaf-nodes}(\mathcal{T}[u \vee v])|.$$

Plugging in this bound in the equation above gives us

$$W(G) \leq (1 + \frac{1-\beta}{\beta}) \cdot \sum_{e=(u,v)\in E} w(e) \cdot |\textbf{leaf-nodes}(\mathcal{T}[u \vee v])| = \frac{1}{\beta} \cdot \textbf{cost}_G(\mathcal{T}),$$

where the final equality is by Eq (1). ■

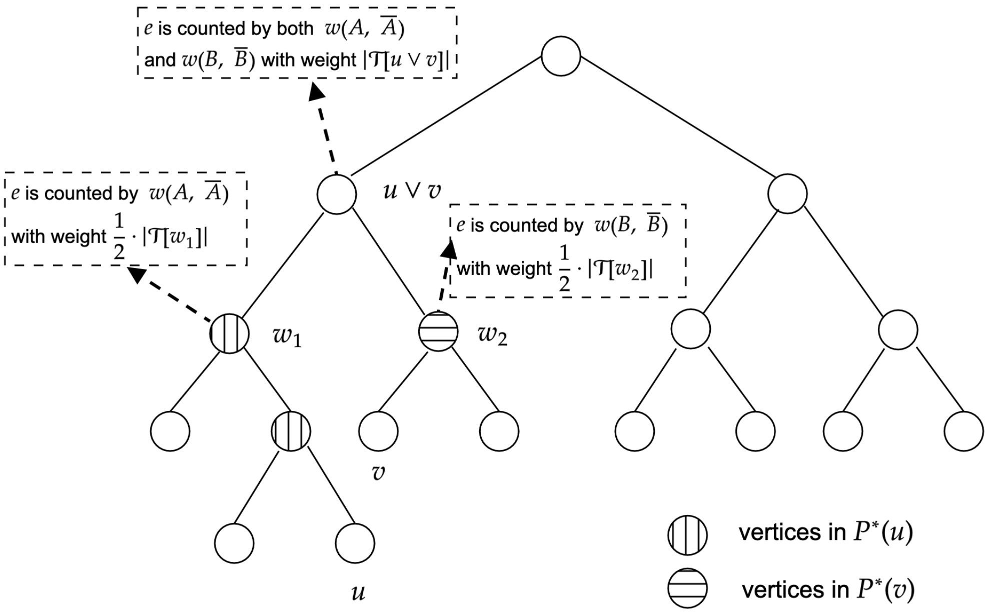

Figure 1: An illustration of the alternative equation for $W(G)$.

C.2. Missing Proofs of Lemma 19 and Proposition 20

If we use the balanced minimum cut to establish a lower bound on the value of the optimal cost, we can obtain a clean proof of $O(\log n)$ approximation. We include this proof in Appendix G. Proving the stronger bound in Lemma 19 requires some more involved techniques first developed by Charikar and Chatziafratis (2017).

Definition 36 *Let $G = (V, E)$ be a graph, $\mathcal{T}$ be a HC-tree of G and $(u, v) \in E$. The* footprint of (u, v) at size t*, denoted by $f^t_{\mathcal{T}}((u, v))$, is defined as follows.*

$$f^t_{\mathcal{T}}((u,v)) = \begin{cases} 0, \text{ if there exists a cluster } C \text{ induced by } \mathcal{T}, \text{ s.t. } |C| \le t \text{ and } |\{u,v\} \cap C| = 1 \\ w(u,v), \text{ otherwise.} \end{cases}$$

We first observe the relationship between the edge footprint and the cost of a hierarchical clustering. Consider an edge (u, v) and let $C = \textbf{leaf-nodes}(\mathcal{T}[u \wedge v])$. Recall that (u, v) contributes a cost of $|C| \cdot w((u, v))$. Thus, the footprint of the edge is equal to its weight for any $t < |C|$. As such, we have the following:

Lemma 37 *Using the assumptions of Definition 36, and assuming that G has n vertices, we have*

$$\frac{1}{3} \cdot \sum_{t=0}^{n} \sum_{(u,v) \in E} f^{(2/3) \cdot t}_{\mathcal{T}}((u,v)) \le \sum_{t=0}^{n} \sum_{(u,v) \in E} f^t_{\mathcal{T}}((u,v)) = \textbf{cost}(\mathcal{T}).$$

Proof We first prove the equality. By the definition of $f^t_{\mathcal{T}}\left((u,v)\right)$, the weight of an edge (u,v) is counted r times where $r = |\textbf{leaf-nodes}(\mathcal{T}[u \wedge v])|$ (note that the first sum starts with $t = 0$).

To prove the inequality, let us assume WLOG that n is a multiplier of 3. Note that the number of terms between $A := \sum_{t=0}^{n}\sum_{(u,v)\in E} f^t_{\mathcal{T}}\left((u,v)\right)$ and $B := \sum_{t=0}^{n}\sum_{(u,v)\in E} f^{(2/3)\cdot t}_{\mathcal{T}}\left((u,v)\right)$ are the same, and we charge B into 3 copies of A. Note that when $(1/3)\cdot t \in \{0, 2, 4, \cdots, \frac{2}{3}\cdot n\}$, we can charge this part of B to $C := \sum_{t=0}^{\frac{2}{3}\cdot n}\sum_{(u,v)\in E} f^t_{\mathcal{T}}\left((u,v)\right)$, which in tern is at most A. For the second case, consider $(2/3)\cdot t \in \{\frac{2}{3}, \frac{8}{3}, \cdots, \frac{2}{3}\cdot n - 2\}$, we introduce another copy of A, and since $f^t_{\mathcal{T}}\left((u,v)\right) < f^{t-2/3}_{\mathcal{T}}\left((u,v)\right)$, this part of B is also upper-bounded by A. Finally, we consider the case $(2/3)\cdot t \in \{\frac{4}{3}, \frac{10}{3}, \cdots, \frac{2}{3}\cdot n - 1\}$. With the same reasoning as above, the quantity of this part of B is again at most A. Therefore, we have $3A \geq B$, as claimed. ■

Note that a result similar to Lemma 37 was first obtained by Charikar and Chatziafratis (2017). However, the subtle difference makes their statement not directly applicable for our purpose.

We say that a β-balanced min cut is a β-balanced cut of minimum weight. We now compare the cost of a tree $\mathcal{T}$ obtained by recursively partitioning the set of nodes using a $\frac{1}{3}$-balanced min cut and an optimal tree $\mathcal{T}^*$. We first show how to upper-bound the cost incurred by each node in Eq (2) by edge footprints.

Lemma 38 *Let $\mathcal{T}^*$ be a tree of cost $\mathsf{OPT}(G)$ and $\mathcal{T}$ be a tree obtained by recursively applying $\frac{1}{3}$-balanced min cut. Let w be an internal node of $\mathcal{T}$, $S = \textbf{leaf-nodes}(\mathcal{T}[w])$ and $(S_1, S_2) = \textbf{cut}(\mathcal{T}[w])$. Denote $r := |S|$, $s := |S_1|$. Then,*

$$r \cdot w(S_1, S_2) \leq 3 \cdot s \cdot \sum_{(u,v)\in S(E)} f^{(2/3)\cdot r}_{\mathcal{T}^*}\left((u,v)\right).$$

Proof Let $S^\star_1$, $S^\star_2$, $\cdots$, $S^\star_k$ be the maximal (w.r.t. inclusion) clusters induced by $\mathcal{T}^*$, which have size at most $\frac{2}{3}\cdot r$ and a nonempty intersection with S. Observe that these clusters are all disjoint.

We claim there exist two sets of indices $L = \{l_1, l_2, \cdots\}$ and $R = \{r_1, r_2, \cdots\}$, such that

- $L \cup R = [k]$;
- If we denote $A := (\cup_{i\in L} S^\star_i) \cap S$ and $B := (\cup_{i\in R} S^\star_i) \cap S$, we have $\max\{A, B\} \leq \frac{2}{3}\cdot r$.

To prove this claim we use the fact that the intersection of each $S^\star_i$ with S is at most $\frac{2}{3}\cdot r$, and the following observation.

Observation 39 *Let $x_1, \ldots, x_k$ be a sequence of positive real numbers, such that $\sum_{i=1}^{k} x_i = 1$, and $x_i \leq \frac{2}{3}$. There exists a sequence $1 \leq j_1 < \ldots < j_l \leq k$, such that $\frac{1}{3} \leq \sum_{i=1}^{l} x_{j_i} \leq \frac{2}{3}$.*

This implies that the cut (A, B) is $\frac{1}{3}$-balanced, so in particular its weight is at least the weight of the minimum $\frac{1}{3}$-balanced cut. We have

$$\begin{aligned} w(S_1, S_2) &\leq w\left(\cup_{i\in L} S^\star_i\right) \cap S, \left(\cup_{i\in R} S^\star_i\right) \cap S) && \text{(by balanced minimum cut)} \\ &\leq \sum_{(u,v)\in S(E)} f^{(2/3)\cdot r}_{\mathcal{T}^*}\left((u,v)\right). && (3) \end{aligned}$$

The second inequality holds, since each edge in the cut $(\cup_{i \in L} S_i^\star) \cap S, (\cup_{i \in R} S_i^\star) \cap S)$ has exactly one endpoint in some $S_i^\star$ (whose size is at most $(2/3) \cdot r$), and thus a nonzero footprint at level $(2/3) \cdot r$

Since the cut (S_1, S_2) is $\frac{1}{3}$-balanced and, by the definition of **cut**, $|S_1| \leq |S_2|$, we have $\frac{r}{3} \leq s$. Hence, we get

$$r \cdot w(S_1, S_2) \leq 3 \cdot s \cdot \sum_{(u,v) \in S(E)} f_{\mathcal{T}^*}^{(2/3) \cdot r}\left((u,v)\right),$$

as claimed. ■

Corollary 40 *Using the assumptions of Lemma 38:*

$$r \cdot w(S_1, S_2) \leq 3 \cdot \sum_{t=|S_2|+1}^{|S_1|+|S_2|} \sum_{(u,v) \in S(E)} f_{\mathcal{T}^*}^{(2/3) \cdot t}\left((u,v)\right)$$

Proof This follows directly from Lemma 38 and since for any i, $f_{\mathcal{T}^*}^{i}\left((u,v)\right) \leq f_{\mathcal{T}^*}^{i-1}\left((u,v)\right)$. ■

Lemma 41 *Let $\mathcal{T}$ be a tree obtained by recursively applying $\frac{1}{3}$-balanced minimum cut. Consider the sum*

$$\sum_{\substack{(S_1, S_2) := \textbf{cut}(\mathcal{T}[w]) \text{ for} \\ \text{internal nodes } w \text{ of } \mathcal{T}}} \sum_{t=|S_2|+1}^{|S_1|+|S_2|} \sum_{(u,v) \in (S_1 \cup S_2)(E)} F(t, u, v),$$

Then, for any $0 \leq t' \leq n$, and any u', v', the term $F(t', u', v')$ appears at most once in the sum.

Proof Clearly, any possible overlap in the terms can only come from two nodes $w \neq w'$, such that $S = \textbf{leaf-nodes}(\mathcal{T}[w])$, $S' = \textbf{leaf-nodes}(\mathcal{T}[w'])$ and $S \cap S' \neq \emptyset$. WLOG we can assume that w is an ancestor of w'.

Denote $(S_1, S_2) := \textbf{cut}(\mathcal{T}[w])$, where $|S_1| \leq |S_2|$. Since S' is either a subset of S_1 or S_2, we have $|S'| \leq |S_2|$. But then, the largest index t we can obtain when we consider all summands corresponding to w' is $|S_2|$. At the same time, the smallest index t corresponding to w is $|S_2| + 1$. ■

Proof of Lemma 19 With Lemma 37, Corollary 40 and Lemma 41 in our hands, now we can establish the approximation ratio of $\mathcal{T}$ that is obtained by recursive $\frac{1}{3}$-balanced min-cut.

$$\sum_{\substack{(S_1, S_2) := \textbf{cut}(\mathcal{T}[u]) \text{ for} \\ \text{internal nodes } u \text{ of } \mathcal{T}}} w(S_1, S_2) \cdot r \leq 3 \cdot \sum_{\substack{(S_1, S_2) := \textbf{cut}(\mathcal{T}[u]) \text{ for} \\ \text{internal nodes } u \text{ of } \mathcal{T}}} \sum_{t=|S_2|+1}^{|S_1|+|S_2|} \sum_{(u,v) \in S(E)} f_{\mathcal{T}^*}^{(2/3) \cdot t}\left((u,v)\right) \quad \text{(By Corollary 40)}$$

$$= 3 \cdot \sum_{t=0}^{n} \sum_{(u,v)\in E} f_{\mathcal{T}^*}^{(2/3)\cdot t}\left((u,v)\right) \qquad \text{(By Lemma 41 and the disjointness)}$$
$$\leq 9 \cdot \textbf{cost}(\mathcal{T}^*). \qquad \text{(By Lemma 37)}$$

■

Proof of Proposition 20 The polynomial-time algorithm is to *recursively apply the* $O(\sqrt{\log n})$ *approximation algorithm for balanced minimum cuts on the subgraphs of* G. Suppose S_1 and S_2 are obtained by applying the $O(\sqrt{\log n})$-approximation of the balanced minimum cut, by changing the line in Eq (3), we have

$$w(S_1, S_2) \leq O(\sqrt{\log n}) \cdot w\left(\left(\cup_{i\in L} S_i^\star\right) \cap S, \left(\cup_{i\in R} S_i^\star\right) \cap S\right)$$
$$\text{(by } O(\sqrt{\log n})\text{-approximation of balanced minimum cut)}$$
$$\leq O(\sqrt{\log n}) \cdot \sum_{(u,v)\in S(E)} f_{\mathcal{T}^*}^{(2/3)\cdot r}\left((u,v)\right). \tag{4}$$

As such, for a tree $\mathcal{T}$ obtained by recursive $O(\sqrt{\log n})$ approximation of the balanced minimum cut, we have

$$\sum_{\substack{(S_1,S_2) := \textbf{cut}(\mathcal{T}[u]) \text{ for} \\ \text{internal nodes } u \text{ of } \mathcal{T}}} w(S_1, S_2) \cdot r \leq O(\sqrt{\log n}) \cdot \sum_{\substack{(S_1,S_2) := \textbf{cut}(\mathcal{T}[u]) \text{ for} \\ \text{internal nodes } u \text{ of } \mathcal{T}}} \sum_{t=|S_2|+1}^{|S_1|+|S_2|} \sum_{(u,v)\in S(E)} f_{\mathcal{T}^*}^{(2/3)\cdot t}\left((u,v)\right)$$
$$\text{(By Eq (4) and the fact that } f_{\mathcal{T}^*}^{i}\left((u,v)\right) \leq f_{\mathcal{T}^*}^{i-1}\left((u,v)\right)\text{)}$$
$$= O(\sqrt{\log n}) \cdot \sum_{t=0}^{n} \sum_{(u,v)\in E} f_{\mathcal{T}^*}^{(2/3)\cdot t}\left((u,v)\right)$$
$$\text{(By Lemma 41 and the disjointness)}$$
$$\leq O(\sqrt{\log n}) \cdot \textbf{cost}(\mathcal{T}^*). \qquad \text{(By Lemma 37)}$$

We now analyze the time complexity. Note that each approximate balanced minimum cut takes polynomial time. Furthermore, there are at most polynomially-many nodes in a HC-tree since there are at most n leaves. Therefore, the algorithm runs in polynomial time. ■

Appendix D. Missing proof of Theorem 6

Proof Let $k = \Theta(\log^c(n))$ for some fixed constant $c \geq 2$ and suppose G is an n-vertex graph from one of the families of graphs in Proposition 7. Using Lemmas 13 to 15, we can infer the following.

Note that in both cases, each of the k-length paths induces a cost of $k \log k + O(k)$. In case one, there are two cycles, and each of them incurs a cost of at least $\frac{n}{16} \cdot \log\left(\frac{n}{8}\right) + O(n)$ (the lower bound side of Lemma 15). As such, the total cost is at least

$$\mathsf{OPT}(G) \geq 2 \cdot \left(\frac{n}{16} \cdot \log\left(\frac{n}{8}\right) + O(n)\right) + \frac{3n}{4k} \cdot \left(k \log k + O(k)\right) \geq \frac{n}{8} \cdot \log n.$$

On the other hand, in case two, each of the length-$2k$ cycles incurs a cost of at most $2k \cdot \log(2k) + O(k)$ (the upper bound side of Lemma 15). Therefore, the total cost is at most

$$\mathsf{OPT}(G) \leq \frac{n}{8k} \cdot (2k \cdot \log(2k) + O(k)) + \frac{3n}{4k} \cdot (k \log k + O(k)) \leq 2n \cdot \log k.$$

As such, any streaming algorithm that can estimate the value of $\mathsf{OPT}(G)$ to within a factor better than $\frac{\log n}{16 \cdot \log k}$ can distinguish between the two cases for G.

Considering the choice of k, any $o\left(\frac{\log n}{\log \log n}\right)$-approximation algorithm would distinguish the graph families of Proposition 7. Suppose the number of passes of the algorithm is $\sqrt{k} = O(\log^{c/2}(n))$. Thus by Proposition 7, we get that the space of the algorithm is

$$\Omega(\frac{1}{p^5} \cdot (n/k)^{1-\gamma \cdot \frac{p}{k}}) = \Omega(\frac{1}{\text{polylog}(n)} \cdot (n/\text{polylog}(n))^{1-\gamma \cdot 1/\text{polylog}(n)}) = \Omega(n/\text{polylog}(n)).$$

As we can set c to be any arbitrary large constant, we obtain that any polylog (n)-pass streaming algorithm for hierarchical clustering requires $\Omega(n/\text{polylog}(n))$ space. ■

Appendix E. Missing proof of Theorem 9

E.1. Step One: The One-vs-Many-Expanders Problem

We first give the formal definition of *One-vs-Many-Expanders* (OvME) problem, and prove its connection to approximating HC. The problem is defined as follows.

Problem 42 (One-vs-Many-Expanders (OvME)) *For $n, k, t \geq 1$, $\mathsf{OvME}_{n,k,t}$ is a communication game between k players $P_1, P_2, \cdots, P_k$. The input is a graph $G = (V, E)$ on n vertices. Additionally, there is a labeling Σ that partitions vertices of V into t equal-sized classes $(\Sigma_1, \ldots, \Sigma_t)$. The labeling Σ is* unknown *to the players. For $i \in [k]$, player P_i is given a matching M_i of size $n/4$. We are promised that the input to belongs to one of the following two classes, chosen uniformly at random:*

- ***Yes**-case: The input matching M_i to every player P_i is chosen uniformly at random over all possible matchings on V.*
- ***No**-case: The input matching M_i to every player P_i is chosen uniformly at random from all matchings that have exactly $n/4t$ edges from each class Σ_j for $j \in [t]$.*

*Starting from P_1, each player sends a message to the next one and the goal is for P_k to determine whether the input is in **Yes**-case or the **No**-case.*

We allow multi-graphs to be created by the definition of Problem 42 so as to not introduce unnecessary correlation between input of players in the Yes-case.

In the following, we first prove that the inputs in $\mathsf{OvME}_{n,k,t}$, with high probability, results in one expander in **Yes**-case and t expanders in **No**-case.

Lemma 43 *In $\mathsf{OvME}_{n,k,t}$, for $k \geq 40\log(n)$, and $t \leq n^{1/2}$, with probability $1 - o(1)$, we have:*

- *A graph G sampled from* **Yes**-*case consists is a single expander with edge expansion* $\Omega(\log n)$ *and* $nk/4$ *edges.*
- *A graph G sampled from* **No**-*case consists of t expanders, each on n/t vertices and $nk/4t$ edges, with edge expansion* $\Omega(\log n)$.

Before proving this lemma, we need to introduce a standard result.

Claim 44 *For any pairs of vertices u, v in G and $i \in [k]$, define an indicator random variable $x_{u,v,i}$ which is 1 iff (u, v) is sampled in M_i. Then, the set of random variables $\{X_{u,v,i}\}_{u,v\in G, i\in[k]}$ are negatively correlated conditioned on Σ in both* **Yes** *and* **No** *cases* $\mathsf{OvME}_{n,k,t}$.

Proof Firstly, conditioned on the **Yes** or **No** case and Σ, the choice of M_i and M_i for $i \neq j \in [k]$ are independent. Thus, we only need to show that for every $i \in [k]$, $\{X_{u,v,i}\}_{u,v\in G}$ are negatively correlated. We prove this for any pairs of random variables and one can inductively prove it for all $\{X_{u,v,i}\}_{u,v\in G}$ for every $i \in [k]$ as well. Consider the **Yes** case first. We have,

$$(\mathbb{E}\left[X_{u',v',i}\right] =) \, \mathbb{E}\left[X_{u,v,i}\right] = \Pr\left((u,v) \in M_i\right) = \frac{n/4}{\binom{n}{2}} = \frac{1}{2\cdot(n+1)},$$

where the probability calculation holds as each matching is of size $n/4$ chosen uniformly at random.

Similarly, we have,

$$\begin{aligned}\mathbb{E}\left[X_{u,v,i}\cdot X_{u',v',i}\right] &= \Pr\big((u,v)\in M_i \wedge (u',v')\in M_i\big)\\ &\leq \frac{(n/4)\cdot(n/4-1)}{\binom{n}{2}\cdot\binom{n-2}{2}}\\ &\text{(the inequality is tight if } \{u,v\}\cap\{u',v'\}=\emptyset \text{ and otherwise the probability is zero)}\\ &< \mathbb{E}\left[X_{u,v}\right]\cdot\mathbb{E}\left[X_{u',v'}\right]. \qquad \text{(by a direct calculation of the bounds)}\end{aligned}$$

By and inductive argument, we can extend this to all subsets of $\{X_{u,v,i}\}_{u,v\in G}$, proving the negative correlation of the variables in this case.

In the **No** case, we can repeat the same argument for each individual class Σ_j for $j \in [t]$ instead. This finalizes the proof. ∎

Proof of Lemma 43 We first prove the result in the **Yes** case. Let us fix any partition $(S, V \setminus S)$ such that $|S| \leq \frac{n}{2}$. Consider the random variables $X_{u,v,i}$ for $i \in [k]$ in Claim 44 for any pairs of vertices $u \in S$ and $v \in V \setminus S$. Define $X_S := \sum_{i=1}^{k}\sum_{u\in S, v\in V\setminus S} X_{u,v}$. We have,

$$\mathbb{E}\left[X_S\right] = \sum_{u\in S, v\in V\setminus S}\sum_{i=1}^{k}\mathbb{E}\left[X_{u,v}\right] \leq |S|\cdot\frac{n}{2}\cdot\frac{k}{2\cdot(n+1)} \leq |S|\cdot 5\log n,$$

as $|S| \leq n/2$, by the bound on expectation in Claim 44, and since $k \geq 40\log n$.

As X_S is a sum of negatively correlated $\{0,1\}$-random variables by Claim 44, we can apply Chernoff bound for negatively correlated random variables (Proposition 29), to have

$$\Pr\left(X_S \leq |S|\cdot\log(n)\right) \leq \exp\left(-4\,|S|\cdot\log(n)\right).$$

As such, one can apply union bound for all partitions with size $|S| := s \geq 1$ as

$$\Pr\left(\exists S \ s.t. \ (|S| = s) \wedge (X_S \leq s \cdot \log(n))\right) \leq \binom{n}{s} \exp\left(-4s \cdot \log(n)\right) \leq n^s \cdot \exp\left(-4s \cdot \log(n)\right) \leq \frac{1}{n^2}.$$

Finally, one can apply union bound for all size $s \leq \frac{n}{2}$, and finalize the statement for the **Yes** case.

For the proof for the **No** case, we first observe that no edge will ever be added between two classes Σ_i and Σ_j, and the edges inside each Σ_i are exactly of the size $\frac{nk}{4t}$. Moreover, distribution of each graph induced on Σ_i matches that of **Yes** case on the whole graph. Thus, we can apply the same argument as before to each Σ_i individually and obtain the same lower bound of $\Omega(\log n)$ on their expansion. This concludes the proof. ■

By Lemma 43, in order to prove Theorem 9, we need to lower bound the communication complexity of $\mathsf{OvME}_{n,k,t}$ for $k = 40\log(n)$ and $t = n^{1/2-\delta}$. It is well-known that the one-way communication lower bound implies a single-pass streaming memory lower bound on the same input distribution: the reduction is to simply let each player run the streaming algorithm and send the memory as the message. Thus, the space of the algorithm would be an upper bound on the communication in the protocol.

E.2. Step Two: The Hidden Labeling Problem (HLP)

To prove a lower bound for OvME, we define an intermediate two-player communication problem.

Problem 45 (*Hidden Labeling Problem* (HLP)) *For $n, t \geq 1$, $\mathsf{HLP}_{n,t}$ is a two player communication game between Alice and Bob. We have a graph $G = (V, E)$, Alice is given a labeling Σ of V into t equal-size classes $(\Sigma_1, \ldots, \Sigma_t)$. Bob is given a single matching M of size $n/4$. We are promised that the input is one of the following two cases chosen uniformly at random:*

- **Yes**-*case: The matching M of Bob is chosen uniformly at random from all matchings on V.*
- **No**-*case: The matching M of Bob is chosen uniformly at random from all matchings that contain exactly $n/4t$ edges from each class Σ_j for $j \in [t]$.*

The goal is for Alice to send a message to Bob, and Bob outputs which case the input belongs to.

Intuitively, if there is a protocol that solves OvME with high probability, it should gain some information about the distribution we used in HLP also that help outperform random guessing. We formalize this as the following lemma in this step.

Lemma 46 *Suppose there exists a (possibly randomized) communication protocol that uses c bits and solves $\mathsf{OvME}_{n,k,t}$ (Problem 42) with probability at least $1/2 + \varepsilon$ for some $\varepsilon > 0$. Then, there exists a deterministic communication protocol that that uses c bits and solves $\mathsf{HLP}_{n,t}$ (Problem 45) correctly with probability at least $1/2 + \varepsilon/k$.*

We prove Lemma 46 by a standard argument. We first use a hybrid argument to show the existence of an 'informative index' among the message between the players over a hybrid distribution. More formally, we define the distributions $\{\mu(f_i)\}_{i=0}^{k}$ as follows:

- For each $\mu(f_i)$, let matchings $M_1, \ldots, M_i$ be sampled from the **No** case of Problem 42, and the latter $k-i$ matchings be sampled from the **Yes** distribution. We have,

In the following, fix a protocol Π_{OvME} that solves OvME with probability at least $1/2+\varepsilon$. We use $\mathtt{msg}(\Pi_{\mathsf{OvME}})$ to denote the random variable for the messages in Π_{OvME}, including the final answer.

Lemma 47 *There exists an* **informative index** $i^* \in [k]$ *such that*

$$\|(\mathtt{msg}(\Pi_{\mathsf{OvME}}) \mid \mu(f_{i^*-1})) - (\mathtt{msg}(\Pi_{\mathsf{OvME}}) \mid \mu(f_{i^*}))\|_{\mathrm{tvd}} \geq \frac{\varepsilon}{2k}.$$

Proof Note that by the definition of our hybrid distribution, the distribution in Problem 42 is $\nu = \frac{1}{2} \cdot (\mu(f_0) + \mu(f_k))$. Hence, by Fact 34, to determine whether a draw of ν is from $\mu(f_0)$ or $\mu(f_k)$ with probability at least $1/2+\varepsilon$, there must be that

$$\|(\mathtt{msg}(\Pi_{\mathsf{OvME}}) \mid \mu(f_0)) - (\mathtt{msg}(\Pi_{\mathsf{OvME}}) \mid \mu(f_k))\|_{\mathrm{tvd}} \geq \frac{\varepsilon}{2}.$$

On the other hand, by the triangle inequality of total variation distance, we have

$$\begin{aligned}&\|(\mathtt{msg}(\Pi_{\mathsf{OvME}}) \mid \mu(f_0)) - (\mathtt{msg}(\Pi_{\mathsf{OvME}}) \mid \mu(f_k))\|_{\mathrm{tvd}}\\ &\leq \sum_{i=1}^{k} \|(\mathtt{msg}(\Pi_{\mathsf{OvME}}) \mid \mu(f_{i-1})) - (\mathtt{msg}(\Pi_{\mathsf{OvME}}) \mid \mu(f_i))\|_{\mathrm{tvd}}.\end{aligned}$$

An averaging argument now concludes the proof. ■

Based on Lemma 47, we can now design a protocol Π_{HLP} for Alice and Bob to gain an advantage of $\Omega(\frac{\varepsilon}{k})$ for HLP. In what follows, we will use $\mathtt{msg}(\Pi_{\mathsf{OvME}})_i$ as the message of player P_i and $\mathtt{msg}(\Pi_{\mathsf{OvME}})_{<i}$ as the messages of players $P_1, \ldots, P_{i-1}$ in Π_{OvME}.

Π_{HLP}: a communication protocol for Theorem 45.

Input:

- Π_{OvME}: a communication protocol for Problem 42 that uses c bits of communication.
- The informative index i^* of Lemma 47 for Π_{OvME}
- (Σ, M): the inputs of Alice and Bob as prescribed in Problem 45.

Alice:

(i) Alice samples the first i^*-1 matchings $M_1, \ldots, M_{i^*-1}$ of OvME as input to players $P_1, \ldots, P_{i^*-1}$ following the **No** distribution conditioned on her input labeling Σ.

(ii) Alice runs Π_{OvME} for the first i^*-1 players and sends $\mathtt{msg}(\Pi_{\mathsf{OvME}})_{<i^*}$ to Bob.

Bob:

(i) Bob picks the input of player P_{i^*} to be the matching M in his input to HLP.

(ii) Bob further samples the inputs to players $P_{i^*+1}, \ldots, P_k$ by sampling the matchings $M_{i+1}, \ldots, M_k$ from the **Yes**-distribution (using the fact that in this case, the matchings are independent of Σ which is unknown to Bob).

(iii) Bob continues running Π_{OvME} on the remaining players using $\mathtt{msg}(\Pi_{\mathsf{OvME}})_{<i^*}$ get the final message $\mathtt{msg}(\Pi_{\mathsf{OvME}})$.

(iv) Bob returns the MLE of $\mathtt{msg}(\Pi_{\mathsf{OvME}})$ between the distributions $\mu(f_{i^*-1})$ and $\mu(f_{i^*})$, i.e., returns **Yes** if $\mu(f_{i^*-1})(\mathtt{msg}(\Pi_{\mathsf{OvME}})) \geq \mu(f_{i^*})(\mathtt{msg}(\Pi_{\mathsf{OvME}}))$ and **No** otherwise.

It is easy to see that Π_{HLP} is a valid communication protocol with c bits of communication. We now prove the correctness of Π_{HLP}.

Claim 48 *The graph G created in Π_{HLP} follows $\mu(f_{i^*-1})$ if (Σ, M) is a* **Yes** *case, and follows $\mu(f_{i^*})$ if (Σ, M) is a* **No** *case.*

Proof Consider the process of drawing from $\mu(f_{i^*-1})$ or $\mu(f_{i^*})$: the first $(i^* - 1)$ coordinates follows the **No** distribution, which is exactly sampled by Alice. The last $(k - i^*)$ coordinates follows the **Yes** distribution, which is exactly sampled by Bob. The i^*-th matching depends on whether (Σ, M) is a **Yes**-case or a No-case, as desired. ∎

Proof of Lemma 46 Consider the distribution $\frac{1}{2} \cdot (\mu(f_{i^*-1}) + \mu(f_{i^*}))$ and note that by Claim 48, this is the distribution of graph G when (Σ, M) is sampled in HLP. By Lemma 47, we have

$$\|(\mathtt{msg}(\Pi_{\mathsf{OvME}}) \mid \mu(f_{i^*-1})) - (\mathtt{msg}(\Pi_{\mathsf{OvME}}) \mid \mu(f_{i^*}))\|_{\mathrm{tvd}} \geq \frac{\varepsilon}{2k}.$$

By Fact 34, we know that Bob can distinguish whether $\mathtt{msg}(\Pi_{\mathsf{OvME}})$ follows $\mu(f_{i^*-1})$ or $\mu(f_{i^*})$ with probability at least $1/2 + \varepsilon/k$, which implies that Π_{HLP} will output the correct answer with the same probability at least.

Finally, note that the protocol Π_{HLP} designed earlier is randomized. However, given that we are measuring success of the protocol against a fixed hard input distribution, we can simply fix its public randomness by an averaging argument and obtain a deterministic protocol with the same probability of success (or equivalently, apply the easy direction of Yao's minimax principle). This concludes the proof. ∎

E.3. Step Three: Decorrelation of HLP

The challenge in analyzing HLP directly is that the edges of Bob are highly correlated. As such, we use another type of hybrid argument to decorrelate these edges. That is, we show that if there is a protocol that solves HLP with advantage $\Omega(\frac{1}{k})$, we can construct a protocol that distinguishes a *single edge* from the **Yes** and **No** cases of HLP, albeit with an advantage which is roughly a factor n smaller. We define the following intermediate problem.

Problem 49 (***Single-Edge Labeling Problem*** ($\mathsf{HLP}^\star$)) *For integer $m \geq 1$, $\mathsf{HLP}^\star_m$ is a two player communication game between Alice and Bob. We have a graph $G = (V, E)$ on m vertices. Alice is given a partitioning of V into two equal-size sets U_0 and U_1. Bob is given a single edge e. We are promised that the input is one of the following two cases chosen uniformly at random:*

- **Yes**-*case: The edge e of Bob is chosen uniformly at random from all pairs of vertices that are between U_0 and U_1.*
- **No**-*case: The edge e of Bob is chosen uniformly at random from all pairs of vertices that are either both belong to U_0 or both to U_1.*

The goal is for Alice to send a message to Bob, and Bob outputs which case the input belongs to.

We now show that a protocol for HLP also implies a protocol for $\mathsf{HLP}^\star$ with non-trivial performance (albeit not measured in terms of the success probability of the protocol). To do so, we need the following definition.

- For any protocol $\Pi_{\mathsf{HLP}^\star}$ of $\mathsf{HLP}^\star$ on (U_0, U_1, e), we define the **advantage** of $\Pi_{\mathsf{HLP}^\star}$ as

  $$\text{advantage}(\Pi_{\mathsf{HLP}^\star}) := \mathop{\mathbb{E}}_{msg} \mathbb{D}(e \mid \mathtt{msg}(\Pi_{\mathsf{HLP}^\star}) = msg, \textbf{Yes} \mid\mid e \mid \mathtt{msg}(\Pi_{\mathsf{HLP}^\star}) = msg, \textbf{No}),$$

  where $\mathtt{msg}(\Pi_{\mathsf{HLP}^\star})$ is the message of Alice to Bob in $\Pi_{\mathsf{HLP}^\star}$ and $\mathbb{D}(\cdot \mid\mid \cdot)$ is the KL-divergence.

Roughly speaking, the advantage of a protocol is a measure of success of the protocol not in terms of probability of outputting the answer, but rather KL-divergence of the distributions of Bob's view of the input between **Yes** and **No** cases, *conditioned* on the message he receives from Alice. We prove that a good protocol for HLP in terms of probability of success implies a protocol for $\mathsf{HLP}^\star$ (on a somewhat smaller instance) with a non-trivial advantage.

Lemma 50 *Let $n, t \geq 1$ be such that $t \leq n^{1/2}$. Suppose there exists a deterministic protocol Π_{HLP} that uses c bits of communication and solves $\mathsf{HLP}_{n,t}$ (Problem 45) with probability at least $1/2 + \varepsilon$ for some $\varepsilon > 1/n$. Then, there also exists a deterministic protocol $\Pi_{\mathsf{HLP}^\star}$ for $\mathsf{HLP}^\star_m$ for some $m = \Theta(n/t)$ (Problem 49) with c bits of communication and* advantage$(\Pi_{\mathsf{HLP}^\star}) \geq \varepsilon^2/2n$.

The proof of this lemma is also based on a hybrid argument, although quite different from that of Lemma 46. For the rest of the proof, fix a protocol Π_{HLP} as in Lemma 50. By Fact 34,

$$\|(M, \mathtt{msg}(\Pi_{\mathsf{HLP}}) \mid \textbf{Yes}) - (M, \mathtt{msg}(\Pi_{\mathsf{HLP}}) \mid \textbf{No})\|_{\text{tvd}} \geq \frac{\varepsilon}{2},$$

since given only $(M, \mathtt{msg}(\Pi_{\mathsf{HLP}}))$, Bob is able to solve HLP with probability of success at least $1/2 + \varepsilon$. By Pinsker's inequality (Fact 33), this implies that

$$\min\{1, \mathbb{D}(M, \mathtt{msg}(\Pi_{\mathsf{HLP}}) \mid \textbf{Yes} \mid\mid M, \mathtt{msg}(\Pi_{\mathsf{HLP}}) \mid \textbf{No})\} \geq \frac{\varepsilon^2}{2}, \tag{5}$$

where we also used the trivial upper bound of 1 on the total variation distance. By the chain rule of KL-divergence (Fact 30), for the LHS of Eq (5), we have,

$$\mathbb{D}(M, \mathtt{msg}(\Pi_{\mathsf{HLP}}) \mid \textbf{Yes} \mid\mid M, \mathtt{msg}(\Pi_{\mathsf{HLP}}) \mid \textbf{No})$$

$$= \mathbb{D}(\texttt{msg}(\Pi_{\mathsf{HLP}}) \mid \textbf{Yes} \mid\mid \texttt{msg}(\Pi_{\mathsf{HLP}}) \mid \textbf{No}) \\ + \mathop{\mathbb{E}}_{msg \mid \textbf{Yes}} \mathbb{D}(M \mid \texttt{msg}(\Pi_{\mathsf{HLP}}) = msg, \textbf{Yes} \mid\mid M \mid \texttt{msg}(\Pi_{\mathsf{HLP}}) = msg, \textbf{No}) \\ = \mathop{\mathbb{E}}_{msg} \mathbb{D}(M \mid \texttt{msg}(\Pi_{\mathsf{HLP}}) = msg, \textbf{Yes} \mid\mid M \mid \texttt{msg}(\Pi_{\mathsf{HLP}}) = msg, \textbf{No}),$$

as the marginal distribution of $\texttt{msg}(\Pi_{\mathsf{HLP}})$ is the same under **Yes** and **No** cases (recall that Alice on her own can only guess the correct answer with probability half).

We denote $M = (e_1, \ldots, e_{n/4})$ where e_i is the i-th edge we sample in the matching M. We further write $M_{<i}$ to denote $e_1, \ldots, e_{i-1}$. Another application of chain rule implies that

$$\mathop{\mathbb{E}}_{msg} \mathbb{D}(M \mid \texttt{msg}(\Pi_{\mathsf{HLP}}) = msg, \textbf{Yes} \mid\mid M \mid \texttt{msg}(\Pi_{\mathsf{HLP}}) = msg, \textbf{No}) \\ = \sum_{i=1}^{n/4} \mathop{\mathbb{E}}_{msg} \mathop{\mathbb{E}}_{M_{<i} \mid msg, \textbf{Yes}} \mathbb{D}(e_i \mid \texttt{msg}(\Pi_{\mathsf{HLP}}) = msg, M_{<i}, \textbf{Yes} \mid\mid e_i \mid \texttt{msg}(\Pi_{\mathsf{HLP}}) = msg, M_{<i}, \textbf{No}).$$

Recall that the distribution of $M_{<i} \mid$ **Yes** is the uniform distribution over all matchings of size $i-1$, independent the input (and thus message) of Alice. We denote this distribution by $\mathcal{U}_{<i}$ (or $\mathcal{U}$ if it is clear from the context). Combining this equation with Eq (5) and an averaging argument implies that there exists an index $i^* \in [n/4]$ such that

$$\mathop{\mathbb{E}}_{M_{<i^*} \sim \mathcal{U}} \mathop{\mathbb{E}}_{msg} \min\{1, \mathbb{D}(e_{i^*} \mid \texttt{msg}(\Pi_{\mathsf{HLP}}) = msg, M_{<i^*}, \textbf{Yes} \mid\mid e_{i^*} \mid \texttt{msg}(\Pi_{\mathsf{HLP}}) = msg, M_{<i^*}, \textbf{No})\} \geq \frac{\varepsilon^2}{2n}. \tag{6}$$

In the rest of the proof, we denote $M_{<i^*}$ and e_{i^*} by M^* and e^* to avoid the clutter. Our goal is now to "massage" the LHS of Eq (6) into a more suitable form for obtaining a protocol for $\mathsf{HLP}^\star$. For any matching M^*, we define:

- $\Sigma(M^*)$ as the choice of classes of vertices of M^*;
- $\mathcal{E}(M^*, \Sigma(M^*))$: the event that the matching M^* matches at most $2n/3t$ vertices from each class Σ_j for $j \in [t]$. Note that this event is fully determined by $(M^*, \Sigma(M^*))$.

We have the following claim that is based on the fact that $\mathcal{E}(M^*, \Sigma(M^*))$ is quite a likely event.

Claim 51 *There exists a choice of $(M^*, \Sigma(M^*))$ such that $\mathcal{E}(M^*, \Sigma(M^*))$ holds and*

$$\mathop{\mathbb{E}}_{msg \mid M^*, \Sigma(M^*)} \mathbb{D}(e^* \mid \texttt{msg}(\Pi_{\mathsf{HLP}}) = msg, M^*, \Sigma(M^*), \textbf{Yes} \mid\mid e^* \mid \texttt{msg}(\Pi_{\mathsf{HLP}}) = msg, M^*, \Sigma(M^*), \textbf{No}) \geq \frac{\varepsilon^2}{4n}.$$

Proof We can write the LHS of Eq (6) as

$$\mathop{\mathbb{E}}_{M^*, msg} \min\{1, \mathbb{D}(e^* \mid \texttt{msg}(\Pi_{\mathsf{HLP}}) = msg, M^*, \textbf{Yes} \mid\mid e^* \mid \texttt{msg}(\Pi_{\mathsf{HLP}}) = msg, M^*, \textbf{No})\} \leq$$

$$\mathop{\mathbb{E}}_{M^*,msg,\Sigma(M^*)} \min\left\{1, \mathbb{D}(e^* \mid \texttt{msg}(\Pi_{\mathsf{HLP}}) = msg, M^*, \Sigma(M^*), \textbf{Yes} \mid\mid e^* \mid \texttt{msg}(\Pi_{\mathsf{HLP}}) = msg, M^*, \Sigma(M^*), \textbf{No})\right\}.$$

(as conditioning cannot decrease the KL-divergence (Theorem 32))

Consider the probability of the event $\mathcal{E}(M^*, \Sigma(M^*))$. Given that M^* is a matching of size at most $n/4$ chosen over $[n]$ uniformly at random in this case (independent of Σ), we have that,

$$\mathbb{E}\,|V(M^*) \cap \Sigma_j| = n/2t,$$

for every $j \in [t]$. Given that $n/t = \omega(\log n)$ in Lemma 50, an application of Chernoff bound for negatively correlated random variables (which holds by Claim 44) plus union bound implies that

$$\Pr\left(\overline{\mathcal{E}(M^*, \Sigma(M^*))}\right) \le \sum_{j=1}^{t} \Pr\left(|V(M^*) \cap \Sigma_j| > \frac{4}{3} \cdot \mathbb{E}\,|V(M^*) \cap \Sigma_j|\right) \ll 1/\text{poly}(n).$$

Thus, we have,

$$\mathop{\mathbb{E}}_{M^*,msg,\Sigma(M^*)} \min\left\{1, \mathbb{D}(e^* \mid \texttt{msg}(\Pi_{\mathsf{HLP}}) = msg, M^*, \Sigma(M^*), \textbf{Yes} \mid\mid e^* \mid \texttt{msg}(\Pi_{\mathsf{HLP}}) = msg, M^*, \Sigma(M^*), \textbf{No})\right\}$$
$$\le \mathop{\mathbb{E}}_{M^*,\Sigma(M^*)\mid\mathcal{E}(M^*,\Sigma(M^*))} \; \mathop{\mathbb{E}}_{msg\mid M^*,\Sigma(M^*)} \mathbb{D}(e^* \mid \texttt{msg}(\Pi_{\mathsf{HLP}}) = msg, M^*, \Sigma(M^*), \textbf{Yes} \mid\mid e^* \mid \texttt{msg}(\Pi_{\mathsf{HLP}}) = msg, M^*, \Sigma(M^*), \textbf{No}) + 1/\text{poly}(n),$$

where the $1/\text{poly}(n)$ term accounts for the contribution of KL-divergence terms whenever the event $\mathcal{E}(M^*, \Sigma(M^*))$ does not happen, given that we always truncate the value of KL-divergence by at most one in the prior terms. This bound, together with the fact that $\varepsilon > 1/n$ in Lemma 50 implies,

$$\mathop{\mathbb{E}}_{M^*,\Sigma(M^*)\mid\mathcal{E}(M^*,\Sigma(M^*))} \; \mathop{\mathbb{E}}_{msg\mid M^*,\Sigma(M^*)} \mathbb{D}(e^* \mid \texttt{msg}(\Pi_{\mathsf{HLP}}) = msg, M^*, \Sigma(M^*), \textbf{Yes} \mid\mid e^* \mid \texttt{msg}(\Pi_{\mathsf{HLP}}) = msg, M^*, \Sigma(M^*), \textbf{No}) \ge \frac{\varepsilon}{4n}.$$

An averaging argument allows us to fix the choice of M^* and conclude the proof. ■

Let us now consider the distribution of underlying variables after we condition on $(M^*, \Sigma(M^*))$ in Claim 51. Define $\Sigma' := (\Sigma'_1, \ldots, \Sigma'_t)$ to be a random variable for the choice of classes for remaining vertices. By the definition of $\mathcal{E}(M^*, \Sigma(M^*))$, we have that $\left|\Sigma'_j\right| \ge n/3t$. We can consider the choice of (e^*, Σ') conditioned on $M^*, \Sigma(M^*)$ as follows[8]:

(i) Sample $\Sigma^* = (\Sigma^*_1, \ldots, \Sigma^*_t)$ each of size exactly $n/3t$ vertices, conditioned on $M^*, \Sigma(M^*)$ and the event that both endpoints of e^* belong to Σ^*. Note that by the definition of $\mathcal{E}(M^*, \Sigma(M^*))$ this is valid as each Σ^*_j is chosen from a set of *at least* $n/3t$ vertices.

8. The following analogy may help provide more intuition for this step: suppose we have two red balls and three green balls and we want to pick one ball uniformly at random from one of the two colors; we can first choose one random red ball and one random green ball, and then pick one of these two balls uniformly at random.

(ii) Sample $j_1, j_2 \in [t]$ uniformly at random.

(iii) Let f_0 be an edge chosen uniformly at random with both endpoints from either only in $\Sigma^*_{j_1}$ or only in $\Sigma^*_{j_2}$. Let f_1 be an edge chosen uniformly at random with one endpoint from $\Sigma^*_{j_1}$ and another from $\Sigma^*_{j_2}$.

(iv) Define $\mu(0)$ as the distribution of f_0 and $\mu(1)$ as the distribution of f_1.

By this construction, we have,

- Distribution of e^* in $(\mathsf{HLP} \mid M^*, \Sigma(M^*), \textbf{Yes})$ is

$$\frac{n+3}{2n+3} \cdot \mu(0) + \frac{n}{2n+3} \cdot \mu(1).$$

- Distribution of e^* in $(\mathsf{HLP} \mid M^*, \Sigma(M^*), \textbf{No})$ is $\mu(0)$.

To see why this is the case, notice that after committing to $(\Sigma^*_1, \ldots, \Sigma^*_t)$ conditioned on e^* being incident on them, to choose e^* in $\mathsf{HLP} \mid M^*, \Sigma(M^*), \textbf{Yes}$, we can first pick two classes uniformly at random, and then pick an edge uniformly at random over these vertices. This way, the edge will have both endpoints inside one of the classes with probability

$$\frac{2 \cdot \binom{n/3}{2}}{\binom{2n/3}{2}} = \frac{2 \cdot (n/3) \cdot (n/3+1)}{(2n/3) \cdot (2n/3+1)} = \frac{n+3}{2n+3}.$$

Thus, picking f_0 with this probability and f_1 with one minus this probability is equivalent to sampling e^* under these conditions. The second case also holds analogously.

We have the following claim using convexity of KL-divergence and an averaging argument.

Claim 52 *There exists a choice of $j_1, j_2 \in [t]$ and $\Sigma^*_{-(j_1,j_2)} := \Sigma - \left\{\Sigma^*_{j_1}, \Sigma^*_{j_2}\right\}$, such that*

$$\mathop{\mathbb{E}}_{msg \mid M^*, \Sigma(M^*), j_1, j_2, \Sigma^*_{-(j_1,j_2)}}$$
$$\mathbb{D}(f_1 \mid \mathtt{msg}(\Pi_{\mathsf{HLP}}) = msg, j_1, j_2, \Sigma^*_{-(j_1,j_2)} \mid\mid f_0 \mid \mathtt{msg}(\Pi_{\mathsf{HLP}}) = msg, j_1, j_2, \Sigma^*_{-(j_1,j_2)}) \geq \frac{\varepsilon^2}{2n}.$$

Proof To avoid clutter, we define $Y := j_1, j_2, \Sigma^*_{-(j_1,j_2)}$. We expand the LHS of Claim 51 as follows:

$$\mathop{\mathbb{E}}_{msg \mid M^*, \Sigma(M^*)}$$
$$\mathbb{D}(e^* \mid \mathtt{msg}(\Pi_{\mathsf{HLP}}) = msg, M^*, \Sigma(M^*), \textbf{Yes} \mid\mid e^* \mid \mathtt{msg}(\Pi_{\mathsf{HLP}}) = msg, M^*, \Sigma(M^*), \textbf{No})$$
$$\leq \mathop{\mathbb{E}}_{Y \mid M^*, \Sigma(M^*)} \quad \mathop{\mathbb{E}}_{msg \mid M^*, \Sigma(M^*), Y}$$
$$\mathbb{D}(e^* \mid \mathtt{msg}(\Pi_{\mathsf{HLP}}) = msg, M^*, Y, \textbf{Yes} \mid\mid e^* \mid \mathtt{msg}(\Pi_{\mathsf{HLP}}) = msg, M^*, Y, \textbf{No})$$

(as conditioning cannot decrease KL-divergence (Fact 32))

$$\leq \mathop{\mathbb{E}}_{Y \mid M^*, \Sigma(M^*)} \quad \mathop{\mathbb{E}}_{msg \mid M^*, \Sigma(M^*), Y}$$

$$\mathbb{D}\Big(\frac{(n+3)\cdot\mu(0)+n\cdot\mu(1)}{2n+3}\mid \texttt{msg}(\Pi_{\mathsf{HLP}})=msg,M^*,Y\;||\;\mu(0)\mid \texttt{msg}(\Pi_{\mathsf{HLP}})=msg,M^*,Y\Big)$$

(by the construction stated above)

$$\leq \mathop{\mathbb{E}}_{Y\mid M^*,\Sigma(M^*)}\;\mathop{\mathbb{E}}_{msg\mid M^*,\Sigma(M^*),Y}$$

$$\frac{1}{2}\cdot\mathbb{D}(\mu(1)\mid \texttt{msg}(\Pi_{\mathsf{HLP}})=msg,M^*,Y\;||\;\mu(0)\mid \texttt{msg}(\Pi_{\mathsf{HLP}})=msg,M^*,j_1,j_2,\Sigma^*_{-(j_1,j_2)}).$$

(by the convexity of KL-divergence (Fact 31) and since $(n/2n+3)<1/2$)

This, combined with the RHS of Claim 51 and expanding the definition of Y implies that

$$\mathop{\mathbb{E}}_{j_1,j_2,\Sigma^*_{-(j_1,j_2)}\mid M^*,\Sigma(M^*)}\;\mathop{\mathbb{E}}_{msg\mid M^*,\Sigma(M^*),Y}$$

$$\mathbb{D}(f_1\mid \texttt{msg}(\Pi_{\mathsf{HLP}})=msg,M^*,j_1,j_2,\Sigma^*_{-(j_1,j_2)}\;||\;f_0\mid \texttt{msg}(\Pi_{\mathsf{HLP}})=msg,M^*,j_1,j_2,\Sigma^*_{-(j_1,j_2)})\geq\frac{\varepsilon^2}{2n},$$

which, together with an averaging argument concludes the proof. ■

We are now ready to design a protocol $\Pi_{\mathsf{HLP}^\star}$ for $\mathsf{HLP}^\star_m$ for $m=2n/3t$ as follows:

$\Pi_{\mathsf{HLP}^\star}$: a communication protocol for Problem 49.

Input:

- Π_{HLP}: a communication protocol for Problem 45 that uses c bits of communication.
- A choice of $M^*,j_1,j_2,\Sigma^*_{-(j_1,j_2)}$ as prescribed by Claims 51 and 52.
- (U_0,U_1,e): the inputs of Alice and Bob as prescribed in Problem 49.

Alice:

(i) Alice sets $\Sigma^*_{j_1}=U_0$ and $\Sigma^*_{j_2}=U_1$ to obtain the entire input of Alice-player in Π_{HLP}.

(ii) She then sends the same exact message as Π_{HLP} on this input to Bob.

Note that as we are only interested in the advantage of the protocol $\Pi_{\mathsf{HLP}^\star}$ (and not its output), Bob has no task in this protocol.

Proof of Lemma 50 By the definition of distributions $\mu(0)$ and $\mu(1)$, as well as the randomness of U_0 and U_1, we have that in the protocol $\Pi_{\mathsf{HLP}^\star}$:

$$\text{dist}(U_0,U_1,e)=\text{dist}(\Sigma_{j_1},\Sigma_{j_2},e^*\mid M^*,j_1,j_2,\Sigma^*_{-(j_1,j_2)}),$$

which implies that

$$\text{dist}(\texttt{msg}(\Pi_{\mathsf{HLP}^\star}))=\text{dist}(\texttt{msg}(\Pi_{\mathsf{HLP}})\mid M^*,j_1,j_2,\Sigma^*_{-(j_1,j_2)}),$$
$$\text{dist}(e\mid \texttt{msg}(\Pi_{\mathsf{HLP}^\star}),\textbf{Yes})=\text{dist}(f_1\mid \texttt{msg}(\Pi_{\mathsf{HLP}}),M^*,j_1,j_2,\Sigma^*_{-(j_1,j_2)})$$

$$\text{dist}(e \mid \texttt{msg}(\Pi_{\textsf{HLP}^\star}), \textbf{No}) = \text{dist}(f_0 \mid \texttt{msg}(\Pi_{\textsf{HLP}}), M^*, j_1, j_2, \Sigma^*_{-(j_1,j_2)})$$

As such,

$$\begin{aligned}\text{advantage}(\Pi_{\textsf{HLP}^\star}) &= \mathop{\mathbb{E}}_{msg \sim \Pi_{\textsf{HLP}^\star}} \mathbb{D}(e \mid \texttt{msg}(\Pi_{\textsf{HLP}^\star}) = msg, \textbf{Yes} \mid\mid e \mid \texttt{msg}(\Pi_{\textsf{HLP}^\star}) = msg, \textbf{No}) \\ &= \mathop{\mathbb{E}}_{msg \sim \texttt{msg}(\Pi_{\textsf{HLP}}) \mid M^*, j_1, j_2, \Sigma^*_{-(j_1,j_2)}} \\ &\quad \mathbb{D}(f_1 \mid \texttt{msg}(\Pi_{\textsf{HLP}}) = msg, j_1, j_2, \Sigma^*_{-(j_1,j_2)} \mid\mid f_0 \mid \texttt{msg}(\Pi_{\textsf{HLP}}) = msg, j_1, j_2, \Sigma^*_{-(j_1,j_2)}),\end{aligned}$$

which is at least $\varepsilon^2/2n$ by Claim 52. This proves the bound on advantage($\Pi_{\textsf{HLP}^\star}$). Given that $\Pi_{\textsf{HLP}^\star}$ only communicates a subset of messages communicated by $\Pi_{\textsf{HLP}}$, we have that $\Pi_{\textsf{HLP}^\star}$ also used at most c bits of communication, concluding the proof (note that $\Pi_{\textsf{HLP}^\star}$ is deterministic as long as $\Pi_{\textsf{HLP}}$ was deterministic). ■

E.4. Step Four: A Lower Bound for HLP*

The last step of the proof is the following lemma that gives a lower bound for HLP*. Recall the notion of advantage of a protocol for HLP* defined in the previous subsection.

Lemma 53 *For any integer $m \geq 1$, any deterministic protocol $\Pi_{\textsf{HLP}^\star}$ for $\textsf{HLP}^\star_m$ with $4\log m \leq c \leq m/4$ bits of communication has*

$$\text{advantage}(\Pi_{\textsf{HLP}^\star}) \leq O(1) \cdot (\frac{c}{m})^2.$$

To prove this lemma, we need some definition. Let us denote the input of Alice with a string $x \in \{0,1\}^m$ with $\|x\|_0 = m/2$ where $x_i = 1$ means the i-th vertex belongs to U_1 and $x_i = 0$ means it is in U_0. The input of Bob can also be seen as a pair (i, j) sampled from $[m]$. Define a random variable $Z = x_i \oplus x_j$. When the input is a **Yes**-case, we have that $Z = 1$ and when the input is a **No**-case, $Z = 0$. We denote the message of Alice by Π in this proof. For a message Π, let $X(\Pi)$ denote the set of inputs x that are mapped to the message Π. By the definition of HLP*, conditioned on a message Π, the input of Alice is chosen uniformly at random from $X(\Pi)$. Define:

$$\text{bias}(\Pi, i, j) := \Pr_{x \sim X(\Pi)}(x_i \oplus x_j = 1) - \Pr_{x \sim X(\Pi)}(x_i \oplus x_j = 0),$$

which also gives that

$$\Pr(Z = 1 \mid \Pi, (i,j)) = \frac{1}{2} + \frac{\text{bias}(\Pi, i, j)}{2} \qquad \text{and} \qquad \Pr(Z = 0 \mid \Pi, (i,j)) = \frac{1}{2} - \frac{\text{bias}(\Pi, i, j)}{2}.$$

The first part of the proof is to relate advantage(HLP*) to the bias of underlying variables.

Claim 54 ("advantage is bounded by squared-biases")

$$\text{advantage}(\textsf{HLP}^\star) = O(1) \cdot \mathop{\mathbb{E}}_{\Pi,(i,j)} \left[\text{bias}(\Pi, i, j)^2\right].$$

Proof By the definition of advantage,

$$\text{advantage}(\Pi_{\mathsf{HLP}^\star}) = \mathop{\mathbb{E}}_{\Pi}\left[\mathbb{D}((i,j) \mid \Pi, \textbf{Yes} \,||\, (i,j) \mid \Pi, \textbf{No})\right]$$
$$= \mathop{\mathbb{E}}_{\Pi}\left[\mathbb{D}((i,j) \mid \Pi, Z=1 \,||\, (i,j) \mid \Pi, Z=0)\right].$$

We now expand this KL-divergence term for any choice of message Π of $\Pi_{\mathsf{HLP}^\star}$:

$$\mathbb{D}((i,j) \mid \Pi, Z=1 \,||\, (i,j) \mid \Pi, Z=0)$$
$$= \sum_{(i,j)} \Pr\left((i,j) \mid \Pi, Z=1\right) \cdot \log\left(\frac{\Pr\left((i,j) \mid \Pi, Z=1\right)}{\Pr\left((i,j) \mid \Pi, Z=0\right)}\right)$$

(by the definition of KL-divergence)

$$= \sum_{(i,j)} \frac{\Pr\left(Z=1 \mid \Pi,(i,j)\right) \cdot \Pr\left((i,j) \mid \Pi\right)}{\Pr\left(Z=1 \mid \Pi\right)} \cdot \log\left(\frac{\Pr\left(Z=1 \mid \Pi,(i,j)\right) \cdot \Pr\left(Z=0 \mid \Pi\right)}{\Pr\left(Z=0 \mid \Pi,(i,j)\right) \cdot \Pr\left(Z=1 \mid \Pi\right)}\right)$$

(by Bayes' rule)

$$= 2 \cdot \sum_{(i,j)} \Pr\left(Z=1 \mid \Pi,(i,j)\right) \cdot \Pr\left((i,j) \mid \Pi\right) \cdot \log\left(\frac{\Pr\left(Z=1 \mid \Pi,(i,j)\right)}{\Pr\left(Z=0 \mid \Pi,(i,j)\right)}\right),$$

as $\Pr\left(Z=0 \mid \Pi\right) = \Pr\left(Z=1 \mid \Pi\right) = 1/2$ since conditioned only on Alice's input/message, the answer, namely, Z, is still uniform.

By continuing the above expansion of KL-divergence, we have,

$$\mathbb{D}((i,j) \mid \Pi, Z=1 \,||\, (i,j) \mid \Pi, Z=0)$$
$$= 2 \cdot \sum_{(i,j)} \Pr\left((i,j) \mid \Pi\right) \cdot \Pr\left(Z=1 \mid \Pi,(i,j)\right) \cdot \log\left(\frac{\Pr\left(Z=1 \mid \Pi,(i,j)\right)}{\Pr\left(Z=0 \mid \Pi,(i,j)\right)}\right)$$
$$= 2 \cdot \sum_{(i,j)} \Pr\left((i,j) \mid \Pi\right) \cdot \Pr\left(Z=1 \mid \Pi,(i,j)\right) \cdot \log\left(\exp\left(\frac{\Pr\left(Z=1 \mid \Pi,(i,j)\right) - \Pr\left(Z=0 \mid \Pi,(i,j)\right)}{\Pr\left(Z=0 \mid \Pi,(i,j)\right)}\right)\right)$$

(as $1+x \leq e^x$ for all $x \in \mathbb{R}$)

$$= 2 \cdot \log e \sum_{(i,j)} \Pr\left((i,j) \mid \Pi\right) \cdot \frac{\Pr\left(Z=1 \mid \Pi,(i,j)\right)}{\Pr\left(Z=0 \mid \Pi,(i,j)\right)} \cdot \left(\Pr\left(Z=1 \mid \Pi,(i,j)\right) - \Pr\left(Z=0 \mid \Pi,(i,j)\right)\right)$$
$$= 2 \cdot \log e \sum_{(i,j)} \Pr\left((i,j) \mid \Pi\right) \cdot \left(\frac{1}{2} + \frac{\text{bias}(\Pi,i,j)}{2}\right)\left(\frac{1}{2} - \frac{\text{bias}(\Pi,i,j)}{2}\right)^{-1} \cdot \text{bias}(\Pi,i,j)$$

(by the definition of $\text{bias}(\Pi,i,j)$)

$$\leq 4 \sum_{(i,j)} \Pr\left((i,j) \mid \Pi\right) \cdot \left(1 + 2 \cdot \text{bias}(\Pi,i,j)\right) \cdot \text{bias}(\Pi,i,j)$$

(as $\log e \leq 2$ and $(1/2+x) \cdot (1/2-x)^{-1} \leq (1+2x)$ for small x)

By bringing back the expectation over the choice of M, we have,

$$\mathop{\mathbb{E}}_{\Pi}\left[\mathbb{D}((i,j) \mid \Pi, Z=1 \,||\, (i,j) \mid \Pi, Z=0)\right] \leq 4 \mathop{\mathbb{E}}_{\Pi}\left[\sum_{(i,j)} \Pr\left((i,j) \mid \Pi\right) \cdot \left(1 + 2 \cdot \text{bias}(\Pi,i,j)\right) \cdot \text{bias}(\Pi,i,j)\right]$$

$$= 8 \mathop{\mathbb{E}}_{\Pi} \mathop{\mathbb{E}}_{(i,j)|\Pi} \left[\text{bias}(\Pi, i, j)^2\right],$$

where we used the fact that

$$\mathop{\mathbb{E}}_{\Pi}\left[\sum_{(i,j)} \Pr\left((i,j) \mid \Pi\right) \cdot \text{bias}(\Pi, i, j)\right] = \mathop{\mathbb{E}}_{\Pi,(i,j)}\left[\Pr\left(Z = 1 \mid \Pi, (i,j)\right)\right] - \mathop{\mathbb{E}}_{M,(i,j)}\left[\Pr\left(Z = 0 \mid \Pi, (i,j)\right)\right]$$
$$= \Pr\left(Z = 1\right) - \Pr\left(Z = 0\right) = 0,$$

by the law of total expectation and since $Z \in \{0, 1\}$ is chosen uniformly with no conditioning. This concludes the proof of the claim. ■

The last main part of the argument is to bound the RHS of Claim 54 using standard tools from Fourier analysis.

Claim 55 *For any message* Π*,*

$$\mathop{\mathbb{E}}_{(i,j)}[\text{bias}(\Pi, i, j)^2] = O(\frac{\log(2^m/\left|X(\Pi)\right|)}{m})^2$$

Proof For any message Π, define:

- $f_\Pi : \{0,1\}^m \to \{0,1\}$ as the characteristic function of $X(\Pi)$.
- $\mathcal{X}_{i,j} : \{0,1\}^m \to \{0,1\}$ as the character function over $[m]$ (see Appendix B.2).

We have that

$$\text{bias}(\Pi, i, j) = \Pr_{x \in X(\Pi)}(x_i \oplus x_j = 1) - \Pr_{x \sim X(\Pi)}(x_i \oplus x_j = 0) = \frac{|\{x : x_i \oplus x_j = 1\}| - |\{x : x_i \oplus x_j = 0\}|}{|X(\Pi)|}$$
$$= -\mathop{\mathbb{E}}_{x \sim X(\Pi)}[\mathcal{X}_{i,j}(x)] = \frac{-1}{|X(\Pi)|} \sum_{x \in \{0,1\}^m} f_\Pi(x) \cdot \mathcal{X}_{i,j}(x) = \frac{-2^m \cdot \hat{f}_\Pi(i,j)}{|X(\Pi)|}$$

(by the definition of Fourier coefficients)

At the same time, by KKL inequality (Proposition 35), for any fixed message $|X(\Pi)|$,

$$\sum_{(i,j)} \text{bias}(\Pi, i, j)^2 = \frac{2^{2m}}{|X(\Pi)|^2} \cdot \sum_{(i,j)} \hat{f}_\Pi(i,j)^2 \leq \gamma^{-2} \cdot \left(\frac{2^m}{|X(\Pi)|}\right)^{\frac{2\gamma}{1+\gamma}}.$$

(by the previous equation and KKL inequality in Proposition 35)

By setting $\gamma = 2 \cdot \log\left(2^m/\left|X(\Pi)\right|\right)^{-1}$, we get that, for any message Π,

$$\mathop{\mathbb{E}}_{(i,j)}[\text{bias}(\Pi, i, j)^2] = O(\frac{\log(2^m/\left|X(\Pi)\right|)}{m})^2,$$

as desired. ■

To conclude the proof of Lemma 53, we need to consider the expectation in RHS of Claim 55 over the choices of Π. To do so, we need a short detour to bound the size of $X(\Pi)$ for a "typical" message Π. Define the following event:

- $\mathcal{E}(\Pi)$: the set of inputs mapped to the message Π satisfies
$$|X(\Pi)| < 2^{-2c} \cdot \binom{m}{m/2};$$
recall that $\binom{m}{m/2}$ is the number of choices for x as input to Alice.

Claim 56 ("typical messages have large pre-image")
$$\Pr\left(\mathcal{E}(\Pi)\right) \leq 2^{-c}.$$

Proof We have,

$$\begin{aligned}
\Pr_{\Pi}(\mathcal{E}(\Pi)) &= \binom{m}{m/2}^{-1} \sum_{x} \mathbb{I}(x \text{ is mapped to some } \Pi \text{ such that } \mathcal{E}(\Pi) \text{ true}) \\
&= \binom{m}{m/2}^{-1} \sum_{\Pi:\mathcal{E}(\Pi) \text{ is true}} |X(\Pi)| \\
&\leq \binom{m}{m/2}^{-1} \sum_{\Pi:\mathcal{E}(\Pi) \text{ is true}} 2^{-2c} \cdot \binom{m}{m/2} && \text{(by the definition of } \mathcal{E}(\Pi)) \\
&\leq \binom{m}{m/2}^{-1} \cdot 2^{c} \cdot 2^{-2c} \cdot \binom{m}{m/2} && \text{(as there are at most } 2^c \text{ messages in total)} \\
&= 2^{-c},
\end{aligned}$$

finalizing the proof. ■

We now have everything to conclude the proof of Lemma 53.

Proof of Lemma 53 For any protocol $\Pi_{\mathsf{HLP}^\star}$, we have,

$$\begin{aligned}
\text{advantage}(\Pi_{\mathsf{HLP}^\star}) &\leq O(1) \cdot \mathop{\mathbb{E}}_{\Pi,(i,j)} \left[\text{bias}(\Pi, i, j)^2\right] && \text{(by Claim 54)} \\
&\leq \Pr\left(\mathcal{E}(\Pi)\right) \cdot 1 + \mathop{\mathbb{E}}_{\Pi|\overline{\mathcal{E}(\Pi)}} \mathop{\mathbb{E}}_{(i,j)|\Pi} \left[\text{bias}(\Pi, i, j)^2\right] \\
& && \text{(the event } \mathcal{E}(\Pi) \text{ is independent of } (i,j) \text{ and is only a function of } \Pi) \\
&\leq 2^{-c} + O(1) \cdot \left(\frac{\log(\frac{2^m}{\binom{m}{m/2} \cdot 2^{-2c}})}{m} \right)^2 \\
& && \text{(by Claim 56 for the first term and Claim 55 for the second)} \\
&\leq 2^{-c} + O(1) \cdot \left(\frac{2c + \log m}{m} \right)^2 \\
& && \text{(as } \tbinom{m}{m/2} \geq 2^m/m \text{ for } m \geq 5 \text{ and } c < m/4) \\
&\leq \frac{1}{m^2} + O(1) \cdot \left(\frac{3c}{m}\right)^2 && \text{(as } c \geq 4 \log m) \\
&\leq O(1) \cdot \left(\frac{c}{m}\right)^2.
\end{aligned}$$

This concludes the proof. ■

E.5. Putting Everything Together: Proof of Theorem 9

We now put all these last four steps together and prove Theorem 9. Suppose towards a contradiction that Theorem 9 is not true.

1. By Lemma 43, a streaming algorithm with $o(n^\delta/\log n)$ space with success probability $2/3$ implies a communication protocol Π_{OvME} for $\mathsf{OvME}_{n,k,t}$ for $k = 40\log n$ and $t = n^{1/2-\delta}$ with $o(n^\delta/\log n)$ communication and $2/3 - o(1)$ probability of success.

2. By Lemma 46, Π_{OvME} for $\mathsf{OvME}_{n,k,t}$ for $k = 40\log n$ and $t = n^{1/2-\delta}$ with $o(n^\delta/\log n)$ communication implies a deterministic protocol Π_{HLP} for $\mathsf{HLP}_{n,t}$ with the same communication and $1/2 + \Omega(1/\log n)$ probability of success (take $\varepsilon = 1/6 - o(1)$ in the lemma).

3. By Lemma 50, Π_{HLP} for $\mathsf{HLP}_{n,t}$ with $o(n^\delta/\log n)$ communication and $1/2 + \Omega(1/\log n)$ success probability, implies a deterministic protocol $\Pi_{\mathsf{HLP}^\star}$ for $\mathsf{HLP}^\star_m$ for $m = \Theta(n/t)$ with same communication and advantage$(\Pi_{\mathsf{HLP}^\star}) = \Omega(1/n\log^2 n)$ (take $\varepsilon = \Omega(1/\log n)$ in the lemma).

4. By Lemma 53, any deterministic protocol $\Pi_{\mathsf{HLP}^\star}$ for $\mathsf{HLP}^\star_m$ of $m = \Theta(n/t) = \Theta(n^{1/2+\delta})$ with communication cost $o(n^\delta/\log n)$ can only have
$$\text{advantage}(\Pi_{\mathsf{HLP}^\star}) = O(1)\cdot\left(\frac{o(n^\delta/\log n)}{\Theta(n^{1/2+\delta})}\right)^2 = o(\frac{1}{n\log^2 n}).$$
(Here, we could apply Lemma 53 as $n^\delta/\log n = o(m)$ and $n^\delta/\log n = \omega(\log m)$.)

But now Lines 3 and 4 contradict each other, finalizing our proof by contradiction of Theorem 9.

Appendix F. Missing proof of Theorem 10

F.1. Warm-up: A Lower Bound for Outputting the Optimal HC Tree

We first show a weaker lower bound for optimal HC algorithms that output the *clustering and the split costs*. More concretely, we give the following lemma.

Lemma 57 (Exact Hierarchical Clustering Lower Bound – Weak version) *Any single-pass streaming algorithm that outputs the optimal hierarchical clustering together with the cost of splitting at each node with probability at least $5/6$ requires a memory of $\Omega(n^2)$ bits.*

To this end, we adopt the following one-way communication game as the machinery.

Problem 58 *Suppose we give Alice a random graph $G = (V, E)$ such that there is an edge between each pair of vertices with probability half. Furthermore, we give Bob a partition of $V = (S \cup \bar{S})$. Alice sends a single message to Bob, and Bob outputs the* exact *cut value of $\delta(S, \bar{S})$ in the end.*

We lower bound the communication complexity of Theorem 58 in the following.

Lemma 59 (Communication Complexity of Problem 58) *Any algorithm that solves Problem 58 with probability at least $4/5$ requires $\Omega(n^2)$ communication.*

We prove the lower bound via reduction from Index. As a reminder (and for completeness), the definition of Index is as follows.

Problem 60 *Alice is given a random N-bit string $x \in \{0,1\}^N$ and Bob is given a random index $i \in [N]$. Alice sends a single message to Bob and Bob outputs x_i.*

It is well-known that any communication protocol with success probability $1/2 + \Omega(1)$ for Index requires $\Omega(N)$ communication Ablayev (1993). We can now prove Theorem 59.

Proof of Lemma 59 Let ALG be a protocol for Problem 58 that uses $o(n^2)$ communication and suppose towards a contradiction that it solves the problem with probability at least $4/5$.

The reduction goes as follows. Given an index $x \in \{0,1\}^N$ for $N = \binom{n}{2}$, Alice creates a random graph $G = (V, E)$ on n vertices such that there is an edge between (u, v) iff $x_{uv} = 1$. By the distribution of x, G is also a random graph as desired in Problem 58. Alice then runs ALG on G and sends its message, together with degrees of all vertices, to Bob. This requires $o(n^2) + O(n \log n) = o(n^2)$ communication.

Bob let $(u, v) \in \binom{V}{2}$ be the vertex pair corresponding to index $i \in [N]$ of his input in Index problem. Bob considers the following two cuts: the cut $S_1 = \{u\}$ and the cut $S_2 = \{u, v\}$ and run ALG for both these cuts separately. The choice of these cuts implies that

- If an edge (u, v) exists, then $\delta(S_2, V \setminus S_2) - \delta(S_1, V \setminus S_1) = \deg(v) - 2$;
- If for (u, v) there is not an edge, then $\delta(S_2, V \setminus S_2) - \delta(S_1, V \setminus S_1) = \deg(v)$.

By the guarantee of ALG, the probability that Bob finds the right answer to both cuts is $\geq 1 - 1/5 - 1/5 = 3/5$. Moreover, Bob can know $\deg v$ exactly as it is sent by Alice separately. Thus, with probability at least $3/5$, Bob can determine whether or not the edge $(u, v) \in E$ which is equivalent to checking if $x_i = 1$ or not, i.e., solve Index. Given the lower bound for Index, we obtain a contradiction, concluding the proof of the lemma. ■

PROOF OF LEMMA 57

We now establish the lower bound in Lemma 57. To this end, we design the following reduction that forms a communication protocol for Problem 58, conditioning on a streaming algorithm `ALG-HC` for HC that outputs the desired information as prescribed in Lemma 57 with a memory of $o(n^2)$ bits and a success probability of at least $\frac{99}{100}$.

The reduction goes as follows. We first create n *additional* vertices, and send them to both Alice and Bob. Alice runs `ALG-HC` with her input graph (with her random edges, the original vertices and the isolated additional vertices), and send the memory of the algorithm to Bob. Bob will perform the following operations from his end: Bob assigns the additional vertices to S and $\bar{S}$ to make the augmented partition balanced (denote them as S' and $\bar{S}'$). Furthermore, Bob adds edges between the vertex pairs inside S' and $\bar{S}'$ to create two complete graphs on his side. Then, Bob receives the message from Alice, and Bob runs `ALG-HC` from Alice's memory and the input he creates. Finally, Bob examines the cost of the first split (denote it as C), and return $C/2n$ as the value of $\delta(S, \bar{S})$.

To prove the above protocol solves Problem 58 with probability at least $\frac{49}{50}$, we only need to show that with high probability, the optimal hierarchical clustering tree will split S and $\bar{S}$. As the first step, we bound the number of edges between S and $\bar{S}$ (and resp. S' and $\bar{S}'$):

Claim 61 *In the graph jointly created by Alice and Bob, the number of edges between* S *and* $\bar{S}$ *(and resp.* S' *and* $\bar{S}'$*) is at most* $\frac{n^2}{4}$ *with probability* $1 - o(1)$.

Proof Let X be the random variable that denotes the number of edges between S and $\bar{S}$. Note that X is only affected by the randomness of Alice's edges and the partition of Bob (and *not* affected by the edges Bob adds). Therefore, we have

$$\mathbb{E}[X] \leq \frac{n^2}{4} \cdot \frac{1}{2} \leq \frac{n^2}{8}.$$

Furthermore, X is a sum of independent indicator random variables. Hence, by Chernoff bound, we have

$$\Pr\left(X \geq \frac{n^2}{4}\right) = \Pr\left(X \geq (1+1) \cdot \mathbb{E}[X]\right) \leq \exp\left(-\frac{1/8 \cdot \mathbb{E}[X]}{2+1}\right) \leq \exp\left(-\frac{1/8 \cdot n}{3}\right) = o(1),$$

as $\mathbb{E}[X] \geq n$. ■

We now show that conditioning on the event of Claim 61, the optimal hierarchical clustering tree always first split the edges between S and $\bar{S}$. To this end, we show the following proposition:

Proposition 62 (Sparsity Split Lemma – Weak Version) *Suppose a graph* G *has 2 cliques* S *and* $\bar{S}$ *such that* $|S| + |\bar{S}| = n$*, and suppose the number of edges between* S *and* $\bar{S}$ *(denote as* $E(S, \bar{S})$*) is at most* $\frac{|S| \cdot |\bar{S}|}{2}$*. Then, the optimal hierarchical clustering tree always first split* S *and* $\bar{S}$.

Proof We prove this by induction. As the base case, suppose when $n = 3$, and $|E(S, \bar{S})| = 1$. Then, to first split the $E(S, \bar{S})$ edge is optimal. Now suppose for $n < n'$ this holds. For the graph with n' vertices, if we first split S and $\bar{S}$, the cost is at most

$$\mathcal{C}_1 = |E(S, \bar{S})| \cdot n + 2 \cdot \frac{1}{3} \cdot (\frac{n^3}{8} - \frac{n}{2}).$$

On the other hand, suppose the optimal clustering starts with splitting some other vertices, one can denote the components after the first split as follows:

- $S_a \cup \bar{S}_a$: let E_a be the set of edges edges between them.
- $S_b \cup \bar{S}_b$: let E_b be the set of edges edges between them.
- The set of edges E_c between 1). S_a and $\bar{S}_b$ and 2). S_b and $\bar{S}_a$.

Note that with the induction hypothesis, the optimal clustering tree will split $S_a \cup \bar{S}_a$ and $S_b \cup \bar{S}_b$ in the way that E_a and E_b are cut first. Therefore, the cost induced by *not* splitting $|E(S, \bar{S})|$ is

$$\mathcal{C}_2 = \left(|S_a| \cdot |S_b| + |\bar{S}_a| \cdot |\bar{S}_b| + |E_c|\right) \cdot n + (|S_a| + |\bar{S}_a|) \cdot |E_a| + (|S_b| + |\bar{S}_b|) \cdot |E_b| \\ + \frac{1}{3}(|S_a|^3 - |S_a| + |\bar{S}_a|^3 - |\bar{S}_a| + |S_b|^3 - |S_b| + |\bar{S}_b|^3 - |\bar{S}_b|),$$

such that $|S_a| + |S_b| = \frac{n}{2}$, $|\bar{S}_a| + |\bar{S}_b| = \frac{n}{2}$, and $|E_a| + |E_b| + |E_c| = |E(S, \bar{S})| \leq \frac{n^2}{4}$. By merging and canceling out different terms, we can show that

$$\begin{aligned}\mathcal{C}_1 - \mathcal{C}_2 = |S_a|^2 \cdot |S_b| + |S_a| \cdot |S_b|^2 + |\bar{S}_a|^2 \cdot |\bar{S}_b| + |\bar{S}_a| \cdot |\bar{S}_b|^2 - (|S_a| + |\bar{S}_a|) \cdot |E_a| - (|S_b| + |\bar{S}_b|) \cdot |E_b| \\ + |E(S, \bar{S})| \cdot n - (|S_a| \cdot |S_b| + |\bar{S}_a| \cdot |\bar{S}_b| + |E_c|) \cdot n.\end{aligned}$$

By switching terms in the above inequality, we note that to show $\mathcal{C}_1 - \mathcal{C}_2 \leq 0$, it suffices to show

$$(n - |S_a| - |\bar{S}_a|) \cdot |E_a| + (n - |S_b| - |\bar{S}_b|) \cdot |E_b| \leq \frac{n}{2} \cdot (|S_a| \cdot |S_b| + |\bar{S}_a| \cdot |\bar{S}_b|).$$

We show the above inequality is indeed true. Note that by our constraints, there is $|E_a| \leq |S_a| \cdot |\bar{S}_a|$ and $|E_b| \leq |S_b| \cdot |\bar{S}_b|$. Hence, we have

$$\begin{aligned}(n - |S_a| - |\bar{S}_a|) \cdot |E_a| + (n - |S_b| - |\bar{S}_b|) \cdot |E_b| &\leq n \cdot |E_a| - (\frac{n}{2} - |S_b|) \cdot |S_a| \cdot |\bar{S}_a| \\ &\quad - (\frac{n}{2} - |\bar{S}_b|) \cdot |S_a| \cdot |\bar{S}_a| + n\,|S_b| \cdot |\bar{S}_b| \\ &\quad - (\frac{n}{2} - |S_a|) \cdot |S_b| \cdot |\bar{S}_b| - (\frac{n}{2} - |\bar{S}_a|) \cdot |S_b| \cdot |\bar{S}_b| \\ &= |S_a| \cdot |S_b| \, (|\bar{S}_a| + |\bar{S}_b|) + |\bar{S}_a| \cdot |\bar{S}_b| \, (|S_a| + |S_b|) \\ &= \frac{n}{2} \cdot (|S_a| \cdot |S_b| + |\bar{S}_a| \cdot |\bar{S}_b|).\end{aligned}$$

That is to say, for graphs in the form as prescribed in Proposition 62, the strategy to first split S and $\bar{S}$ results in the minimum cost. Therefore, the optimal HC tree must split S and $\bar{S}$ first. ■

We can now finalize the proof of Lemma 57. By Claim 61, with probability $1 - o(1)$, the number of edges between S' and $\bar{S}'$ created by Alice and Bob satisfies the condition as in Proposition 62. Moreover, although the joint graph has some multi-edges on the top of the complete graph, it does not change the order of split. Therefore, we can apply Proposition 62 to argue that the edges between S' and $\bar{S}'$ are those to be first split. Finally, the failure probability is bounded by a union bound over the event of Claim 61 not happening and the event that the algorithm fails, which is at most $o(1) + 1/6 < 1/5$. The lower bound now follows from Lemma 59.

F.2. A Lower Bound for Outputting the Optimal Value of HC

We now proceed to the proof of the main result of this section. Similar to the proof of Lemma 57, here we give the following communication game to reduce the hardness from.

Problem 63 *Suppose we give Alice a random bipartite graph $G = (L \cup R, E)$ such that for every pair of vertices $(u, v) \in L \times R$, there is*

$$\begin{cases}(u, v) \in E, & \text{w.p. } \frac{1}{2}; \\ (u, v) \notin E, & \text{w.p. } \frac{1}{2}.\end{cases}$$

Furthermore, we give Bob an index of vertex pair $(i, j) \in L \times R$. Alice is allowed to send a message to Bob once*, and one of the two players has to output if (i, j) is an edge.*

The rest of this section is to prove Theorem 10 in steps.

STEP 1: COMPLEXITY OF PROBLEM 63

Intuitively, Problem 63 answers in the same way of INDEX if we treat each vertex pair as an entry in the array of INDEX. We now formalize this complexity result to show that it requires $\Omega(n^2)$ bits to solve Problem 63.

Lemma 64 (Communication Complexity of Problem 63) *Any algorithm that solves Problem 63 with probability at least $\frac{49}{50}$ requires a communication complexity of $\Omega(n^2)$ bits.*

Proof Again, we are going to design a reduction to use the the complexity of INDEX. Suppose we have a streaming algorithm ALG that solves Problem 63 with probability at least $\frac{49}{50}$. We use this to design a protocol that solves INDEX.

The protocol is as follows. Alice constructs a random graph with n vertices such that $N = \frac{n^2}{4}$, i.e. every possible vertex pair for a bipartite graph with $|L| = |R| = \frac{n}{2}$. Bob is given the same vertices, and he transform his index i^* to the corresponding index of the vertex pair (u, v). Alice runs ALG from her end, send the memory to Bob; Bob runs ALG conditioning on Alice's message, and output 0 if $(u, v) \notin E$, and 1 otherwise.

It is straightforward to see that the index value exactly corresponds to the existence of the edge. Therefore, the protocol succeeds with the same probability of ALG. This implies any such streaming algorithm ALG has to use a memory of $\Omega(N) = \Omega(n^2)$ bits. ■

STEP 2: A REDUCTION TO HIERARCHICAL CLUSTERING

We now proceed to the reduction from Problem 63 to hierarchical clustering. Given a streaming hierarchical clustering algorithm ALG, we can make a protocol for Problem 63 as follows:

PORT: a communication protocol for Problem 63.

Input: ALG – a streaming algorithm that outputs the optimal hierarchical clustering for any signed complete graph with probability at least $\frac{99}{100}$.

The Construction: Alice and Bob construct a graph $G = (V, E)$ as follows.

(i) Both Alice and Bob are given $n = 16N$ vertices.

(ii) Alice constructs the random bipartite graph on $2N$ vertices ($G' = (L \cup R, E)$, $|L| = |R| = N$) as in Problem 63; Bob holds an index $i^* = (u, v) \in [N^2]$.

(iii) Bob adds edges to G in the following manner:

(a) Bob connects u with a set of $4N-1$ vertices (call them $\tilde{S}_1$), and make $S_1 = \{u\} \cup \tilde{S}_1$ a clique.

(b) In the same manner, Bob connects v with another set of $4N - 1$ vertices (call them $\tilde{S}_2$), and make $S_2 = \{v\} \cup \tilde{S}_2$ a clique.

(c) Bob connects every vertex in L *except* u with a set of $3N + 1$ vertices (call them $\tilde{S}_3$), and make $S_3 = L \setminus \{u\} \cup \tilde{S}_3$.

(d) In the same manner, Bob connects every vertex in R *except* v with another set of $3N+1$ vertices (call them $\tilde{S}_4$), and make $S_4 = R \setminus \{v\} \cup \tilde{S}_4$.

(iv) We emphasize that $\tilde{S}_1$, $\tilde{S}_2$, $\tilde{S}_3$, and $\tilde{S}_4$ are disjoint, and Bob can pick the vertices based on the lexicographical orders.

The Message and Output:

(i) Alice runs ALG, and sends the memory of the algorithm as the message to Bob.

(ii) Alice further sends the degrees of the first $2N$ vertex to Bob (with $\tilde{O}(N)$ bits).

(iii) Bob runs ALG based on Alice's message, and outputs based on the cost:

(a) Assume w.log. that $\deg(u) < \deg(v)$.

(b) If the cost equals to

$$\begin{aligned}\textbf{cost} &= \deg(u)\cdot 16N + (\deg(v)-1)\cdot 12N \\ &\quad + \frac{1}{2}\cdot \sum_{w\neq u,v} \deg(w)\cdot 8N + \frac{4}{3}\left((4N)^3 - 4N\right),\end{aligned}$$

then return $(u,v)\in E$.

(c) Otherwise, if the cost equals to

$$\begin{aligned}\textbf{cost} &= \deg(u)\cdot 16N + \deg(v)\cdot 12N \\ &\quad + \frac{1}{2}\cdot \sum_{w\neq u,v} \deg(w)\cdot 8N + \frac{4}{3}\left((4N)^3 - 4N\right),\end{aligned}$$

then return $(u,v)\notin E$.

(d) Otherwise, return FAIL.

An illustration of the constructed graph G can be found as the left plot of Figure 2. It is straightforward to see that the reduction does not increase the communication complexity as long as the memory of ALG is $\Omega(n)$. As such, our task now is to prove that with high constant probability, the optimal cost agrees with the desired value. To this end, we introduce the following proposition which characterizes the optimal tree on a graph constructed by Alice and Bob.

Proposition 65 (Sparsity Split Lemma – Strong Version) *Suppose a graph G has 4 cliques S_1, S_2, S_3 and S_4 such that $|S_1| = |S_2| = |S_3| = |S_4| = s$, and suppose there are only 4 set of edges between them: $E(S_1,S_2)$, $E(S_2,S_3)$, $E(S_3,S_4)$ and $E(S_1,S_4)$. Furthermore, assume w.log. that $|E(S_1,S_2)| + |E(S_1,S_4)| \leq |E(S_2,S_3)| \leq |E(S_3,S_4)|$, the edges of G be with the following properties:*

a). $|E(S_1,S_2)| \leq 1$, $1 \leq |E(S_1,S_4)| \leq |E(S_2,S_3)| \leq \frac{3}{8}\cdot s$.

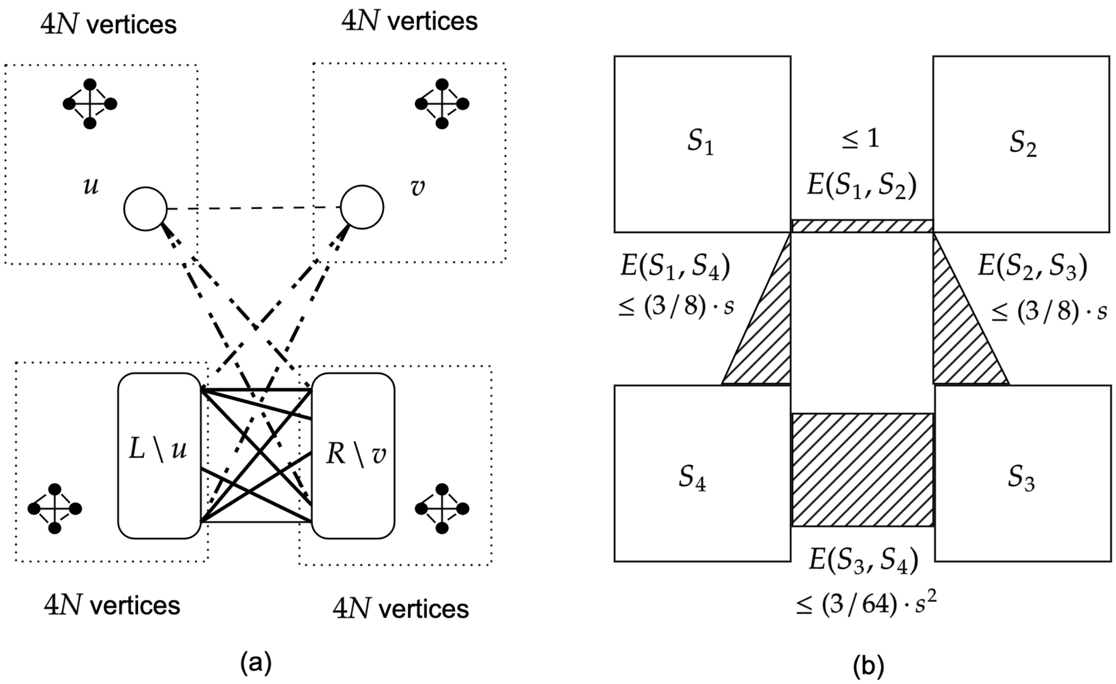

Figure 2: An illustration of the graph constructed by Alice and Bob (a) and the desired graph property of Proposition 65 (b). In (a), the bold solid lines are the random edges added by Alice, minus the edges indent on u and v. The bold dashed lines are the random edges indent on u and v, minus the (possible) edge (u, v) itself. And finally, the dashed line indicates the (possible) edge between u and v. Bob puts the four parts into four components with $4N$ vertices each, and add edges in the components to make them cliques. In (b), the edges of a graph are denoted as the shaded areas. With high constant probability (over the randomness of Alice), the graph G constructed in (a) satisfies the conditions illustrated in (b).

b). Among all vertices $v \in S_1$ (resp. $v \in S_2$), only a single *vertex v has neighbors $u \notin S_1$ (resp. neighbors $u \notin S_2$).*

c). $\frac{1}{64} \cdot s^2 \leq |E(S_3, S_4)| \leq \frac{3}{64} \cdot s^2$. Furthermore, for any $S_3' \subseteq S_3$ and $S_4' \subseteq S_4$, $|E(S_3', S_4')| \leq |S_3'| \cdot |S_4'|$.

An illustration of such a graph G can be found in the right column of Figure 2. Then, the optimal hierarchical clustering tree on G follows the below pattern:

1. The first split separates S_1 from the rest of the graph.

2. *The second split separates S_2 from the rest of the graph.*

3. *The third split separates S_3 and S_4.*

4. *Each clique is clustered by the induced hierarchical clustering tree after it is separated from the rest of the graph.*

In other words, the hierarchical clustering tree is as Figure 3.

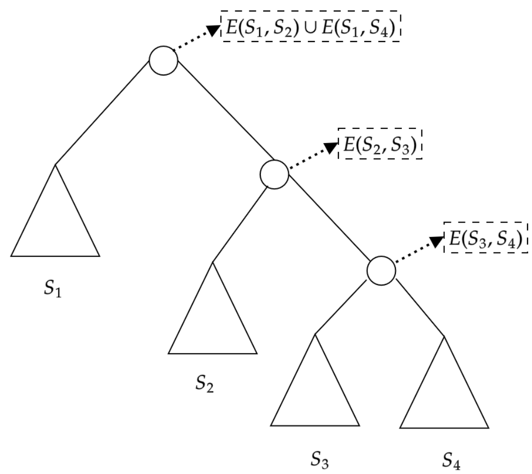

Figure 3: An illustration of the hierarchical clustering tree as described in Proposition 65.

We defer the proof of Proposition 65 to the next step. Conditioning on the statement of Proposition 65, we can show the correctness of PROT. We first show that with high probability, the graph G constructed by Alice and Bob satisfies the conditions required by Proposition 65. More formally, we have

Claim 66 *With probability at least $\frac{99}{100}$, the graph G created by Alice and Bob satisfies the conditions prescribed in Proposition 65 for sufficiently large N.*

Proof We first verify property b).. Note that by our construction, all the edges that go outside S_1 are indent on u, and all edges that go outside S_2 are indent on v. Therefore, property b). holds deterministically.

We now turn to properties a). and c).. For property a)., note that $E(S_1, S_2)$ can only contain the (possible) edge (u, v), which means $|E(S_1, S_2)| \leq 1$ always holds. Let $s = 4N$ as constructed by Alice and Bob. For both $E(S_1, S_4)$ and $E(S_2, S_3)$, the expectation of there size is

$$\mathbb{E}\left[|E(S_1, S_4)|\right] = \mathbb{E}\left[|E(S_2, S_3)|\right] = \frac{1}{2} \cdot (\frac{s}{4} - 1) = \frac{s}{8} - \frac{1}{2}.$$

Therefore, by a Chernoff bound argument, we can show that with probability at least $1 - 2^{-O(s)}$, there are $|E(S_1, S_4)| \leq \frac{3}{8} \cdot s$ and $|E(S_2, S_3)| \leq \frac{3}{8} \cdot s$. With a sufficiently large s, this probability is at least $\frac{199}{200}$.

Finally, for property c)., note that the expected number of edges between S_3 and S_4 is

$$\mathbb{E}\left[|E(S_3, S_4)|\right] = \frac{1}{2} \cdot (\frac{s}{4} - 1)^2 = \frac{s^2}{32} - \frac{s}{4} + \frac{1}{2}.$$

Hence, by a Chernoff bound argument, with probability at least $1 - 2^{-O(s)}$, there is $\frac{1}{64} \cdot s^2 \leq |E(S_3, S_4)| \leq \frac{3}{64} \cdot s^2$. The probability is at least $\frac{199}{200}$ for sufficiently large s. Furthermore, since the graph is simple, the second statement of property c). trivially holds.

A union bound over the failure probability of the above events gives us the desired conclusion. ■

With Proposition 65 and Claim 66, we can establish the correctness of PROT as follows.

Lemma 67 *PROT solves Problem 63 correctly with probability at least* $\frac{49}{50}$.

Proof Conditioning on the event of Claim 66, the optimal hierarchical clustering tree for G follows the pattern prescribed by Proposition 65. Therefore, if $(u, v) \in E$, which means $E(S_1, S_2) = 1$, the optimal tree will first split all the edges on u (which cost $\deg(u)$ edges), then split the remaining edges on v (which cost $\deg(v) - 1$ edges). On the other hand, if $(u, v) \notin E$, which means $E(S_1, S_2) = 0$, the optimal tree will first split all the edges on u (which cost $\deg(u)$ edges), then split the all the edges on v (which cost $\deg(v)$ edges). The remaining part of the splits are the same, and it always confirms to the value described in the reduction. Hence, conditioning on the success of ALG, PROT can correctly distinguish if the edge (u, v) exist.

The failure probability for PROT is at most the union bound over the failure probability of the event of Claim 66 and the failure probability of ALG, which is at most $\frac{1}{50}$. ■

Proof of Theorem 10 Since PROT solves Problem 63 with probability at least $\frac{49}{50}$, by Lemma 64, PROT must send $\Omega(N^2)$ bits. Furthermore, by the reduction, we observe that $n = 16N$. Hence, the message of Alice must be of size at least $\Omega((n/16)^2) = \Omega(n^2)$, which implies the memory of such ALG has to be $\Omega(n^2)$. ■

The rest of our task is to prove Proposition 65.

STEP 3: PROOF OF PROPOSITION 65

The proof of Proposition 65 shares a similar idea to the proof of Proposition 62, albeit the process becomes much more involved. On the high-level, we prove this result in the following steps:

- We first show that conditioning the optimal clustering tree first split the edge *between* cliques, then the optimal strategy is to follow the splits in Proposition 65. This reduces our task to proving the optimal tree always first splits the edges between the cliques.
- The desired statement now is very similar to Proposition 62. However, we need more care in this proof: since a clustering tree splits multiple cliques *in order*, directly applying the inductive proof as in Proposition 62 will create too many cases to handle. Therefore, we instead establish our argument in two steps. The first step is to show that conditioning on the inductive hypothesis, a optimal hierarchical clustering tree will *not* start the edge cuts from

S_1 or S_2. This step relies on the fact that there is only a single vertex inside S_1 and S_2 that connects to vertices outside, and any optimal tree that first splits the clique edges inside has to *entirely* cut the clique edges. We show that this leads to sub-optimal costs.

- The only concern now is the hierarchical clustering tree may start split from the edges inside S_3 or S_4. Since the pattern of split is now controlled, we can employ an inductive argument similar to the proof of Proposition 62, and show that the graph will *not* start split from edges inside S_1 and S_2.
- Finally, we still have to control the behavior of the hierarchical clustering tree *after* the first cut. This part is straightforward: for the subgraph with 3 cliques, we can repeat the argument for 4 cliques. And after we get the subgraph of 2 cliques, we can employ Proposition 62 to get the desired split pattern.

We now formalize the above intuitions. We start with introducing the lemma that controls the behavior of the clustering tree if we restrict the cut to first split edges between cliques.

Lemma 68 *Let G be a graph as prescribed in Proposition 65. For any clustering tree $\mathcal{T}$, if its first 2 cuts are restricted to the edges among $E(S_1, S_2) \cup E(S_2, S_3) \cup E(S_1, S_4) \cup E(S_3, S_4)$, then the optimal cost is induced by the following order: first cut $E(S_1, S_4) \cup E(S_1, S_2)$, then cut $E(S_2, S_3)$.*

Proof Note that if the first two cuts of $\mathcal{T}$ is restricted to edges among $E(S_1, S_2) \cup E(S_2, S_3) \cup E(S_1, S_4) \cup E(S_3, S_4)$, then the third cut should also be among $E(S_1, S_2) \cup E(S_2, S_3) \cup E(S_1, S_4) \cup E(S_3, S_4)$ by Proposition 62. Therefore, the cost of such hierarchical clustering trees is can be characterized as

$$\mathcal{C} = |E_1| \cdot 4s + |E_2| \cdot 3s + |E_3| \cdot 2s + \frac{4}{3} \cdot (s^3 - s),$$

where E_1, E_2 and E_3 are the set of edges to be split in the first, the second, and the third cuts that separates the graph into disconnected components. As a result, it is easy to observe that the minimizer of the cost is attained by always splitting the edges with smaller weights. With the graph described in Proposition 65, it means to first split $E(S_1, S_4) \cup E(S_1, S_2)$, then split (S_2, S_3). ■

We now proceed to the next step, which aims to show that the cuts never start with a cut that splits the clique edges inside S_1 or S_2. More formally, we have

Lemma 69 *Let G be a graph as prescribed in Proposition 65, and $\mathcal{T}$ be the optimal clustering tree of G. Suppose for such a graph G with sizes less than $4s$ (the sizes of S_i's are not necessarily equal), the optimal clustering tree always restrict the first 2 cuts among $E(S_1, S_2) \cup E(S_2, S_3) \cup E(S_1, S_4) \cup E(S_3, S_4)$. Then, for the first cut $G \rightarrow (A, B)$ of $\mathcal{T}$, there is either $S_1 \subseteq A$ or $S_1 \subseteq B$. The same statement holds for S_2.*

Furthermore, if we remove S_1 and all edges indent on S_1 and obtain an induced subgraph G', and let $\mathcal{T}'$ be the optimal tree on G'. Then, for the first cut $G' \rightarrow (A', B')$ of $\mathcal{T}'$, there is either $S_2 \subseteq A'$ or $S_2 \subseteq B'$.

Proof We first observe that there is only one vertex with non-clique edges in S_1 (resp. S_2); therefore, if an optimal tree starts the first split that involves clique edges in S_1 and S_2, it must be *entirely* inside the clique, as the optimal tree never splits the graph into more than two disconnected components. The same holds for S_2.

We now show that restricting the first cut to clique edges inside S_1 or S_2 is sub-optimal. To see this, let a tree $\mathcal{T}'$ be a tree that first splits clique edges in S_1 to induce $S_1 \to (S_1^{in}, S_1^{out})$. Based on the assumption, the optimal tree of the subgraph $G \setminus S_1^{out}$ will restrict its first two cuts among edges of $E(S_1^{in}, S_2) \cup E(S_2, S_3) \cup E(S_1^{in}, S_4) \cup E(S_3, S_4)$. As such, comparing with an optimal tree $\mathcal{T}$ that restrict its first two splits among the edges of $E(S_1, S_2) \cup E(S_2, S_3) \cup E(S_1, S_4) \cup E(S_3, S_4)$, the cost of $\mathcal{T}'$ has the following changes:

- An extra cost of $\left|S_1^{in}\right| \cdot \left|S_1^{out}\right| \cdot 4s$ for the first cut.
- A decreased cost of at most $3s$ multiplicative factor for each of the edges in $E(S_1, S_2) \cup E(S_2, S_3) \cup E(S_1, S_4)$. Hence, this part of decreased cost is at most

$$|E(S_1, S_2) \cup E(S_2, S_3) \cup E(S_1, S_4)| \cdot 3s \leq \frac{9}{4} \cdot s^2.$$

- A decreased cost between splitting S_1 (where the cost is $\frac{1}{3} \cdot (s^3 - s)$) and the cost of splitting S_1^{in} and S_1^{out} separately. This part is at most $\left|S_1^{in}\right|^2 \cdot \left|S_1^{out}\right| + \left|S_1^{in}\right| \cdot \left|S_1^{out}\right|^2 = \left|S_1^{in}\right| \cdot \left|S_1^{out}\right| \cdot s$.

As such, the gap between the costs of $\mathcal{T}'$ and $\mathcal{T}$ is at least

$$\textbf{cost}(\mathcal{T}') - \textbf{cost}(\mathcal{T}) \geq \left|S_1^{in}\right| \cdot \left|S_1^{out}\right| \cdot 4s - \frac{9}{4} \cdot s^2 - \left|S_1^{in}\right| \cdot \left|S_1^{out}\right| \cdot s > 0. \qquad (\left|S_1^{in}\right| \cdot \left|S_1^{out}\right| \geq s - 1)$$

Therefore, such a $\mathcal{T}'$ cannot be an optimal tree on G. ■

We emphasize that the only condition for Lemma 69 to hold is the behavior of the graph in this family with size less than $4s$, which is crucial in our inductive argument of the proof of Proposition 65, established as follows.

Lemma 70 *Let G be a graph as prescribed in Proposition 65 (the sizes of S_i's are not necessarily equal), and let $\mathcal{T}$ be the optimal tree whose first 2 cuts are restricted to the edges among $E(S_1, S_2) \cup E(S_2, S_3) \cup E(S_1, S_4) \cup E(S_3, S_4)$, and let $\mathcal{T}'$ be the optimal tree that whose first cut involved the clique edges. Then, we have* $\textbf{cost}(\mathcal{T}) < \textbf{cost}(\mathcal{T}')$.

Proof We first prove that a tree $\mathcal{T}$ whose *first* cut is restricted to $E(S_1, S_2) \cup E(S_2, S_3) \cup E(S_1, S_4) \cup E(S_3, S_4)$ induces a smaller optimal cost than a tree $\mathcal{T}'$ that first splits edges inside cliques. We can prove this by induction. For the base case, consider all S_i to be single vertices, and there is no clique edges. As such, the statement trivially holds.

For the induction step, suppose the statement holds on such a G with size less than $4s$ (and the sizes of the cliques are not necessarily equal). By Lemma 69, we know that the first cut will *not* start from edges inside S_1 or S_2. Now when $|S_3| = |S_4| = s$, if a clustering tree $\mathcal{T}$ do *not* first split from edges inside S_3 and S_4, the optimal cost is at most

$$\mathcal{C}_1 = (|E(S_1, S_2)| + |E(S_1, S_4)|) \cdot 4s + |E(S_2, S_3)| \cdot 3s + |E(S_2, S_3)| \cdot 2s + \frac{4}{3} \cdot (s^3 - s).$$

On the other hand, suppose a clustering tree $\mathcal{T}'$ starts with splitting edges that involve the clique edges of S_3 and S_4. One can denote the components after the first split as follows:

- $S_1 \cup S_2 \cup S_3^{in} \cup S_4^{in}$: let $E^{in}(S_3, S_4)$ be the set of edges between S_3^{in} and S_4^{in} after the split.
- $S_3^{out} \cup S_4^{out}$: let $E^{out}(S_3, S_4)$ be the set of edges edges between S_3^{out} and S_4^{out}.
- The set of edges $E^{cross} := E^{cross}(S_3, S_4) \cap E^{cross}(S_1, S_4) \cup E^{cross}(S_2, S_3)$ that split to separate S_3^{in} from S_4^{out} and to separate S_4^{in} from S_3^{out}, plus the edges among $E(S_1, S_4)$ that have one vertex in S_4^{out} and the edges among $E(S_2, S_3)$ that have one vertex in S_3^{out}.

We can apply the induction hypothesis such that the subgraphs of $S_1 \cup S_2 \cup S_3^{in} \cup S_4^{in}$ will *not* start with cutting edges inside S_3^{in} and S_4^{in}. Also, not that the optimal tree *unconditionally* will *not* cut the clique edges of S_1 and S_2.

Assuming $\left|E(S_1, S_4^{in})\right| + \left|E(S_2, S_3^{in})\right| \leq \left|E(S_3^{in}, S_4^{in})\right|$, the order of split on the subgraph $S_1 \cup S_2 \cup S_3^{in} \cup S_4^{in}$ does *not* change. As such, the optimal cost induced by this strategy is

$$\begin{aligned}
\mathcal{C}_2 = &\left(\left|S_3^{in}\right| \cdot \left|S_3^{out}\right| + \left|S_4^{in}\right| \cdot \left|S_4^{out}\right| + \left|E^{cross}(S_3, S_4)\right| + \left|E^{cross}(S_1, S_4)\right| + \left|E^{cross}(S_2, S_3)\right|\right) \cdot 4s \\
&+ \left(\left|E(S_1, S_2)\right| + \left|E(S_1, S_4)\right| - \left|E^{cross}(S_1, S_4)\right|\right) \cdot (4s - \left|S_3^{out}\right| - \left|S_4^{out}\right|) \\
&+ \left(\left|E(S_2, S_3)\right| - \left|E^{cross}(S_2, S_3)\right|\right) \cdot (3s - \left|S_3^{out}\right| - \left|S_4^{out}\right|) \\
&+ \left|E^{in}(S_3, S_4)\right| \cdot (\left|S_3^{in}\right| + \left|S_4^{in}\right|) + \left|E^{out}(S_3, S_4)\right| \cdot (\left|S_3^{out}\right| + \left|S_4^{out}\right|) \\
&+ \frac{2}{3}(s^3 - s) + \frac{1}{3}(\left|S_3^{in}\right|^3 + \left|S_4^{in}\right|^3 + \left|S_3^{out}\right|^3 + \left|S_4^{out}\right|^3 - 2s),
\end{aligned}$$

such that the conditions prescribed in Proposition 65 are satisfied. A lower bound of $\mathcal{C}_2$ can be obtained by ignoring the higher cost of $E^{cross}(S_1, S_4)$ and $E^{cross}(S_2, S_3)$:

$$\begin{aligned}
\mathcal{C}_2 \geq &\left(\left|S_3^{in}\right| \cdot \left|S_3^{out}\right| + \left|S_4^{in}\right| \cdot \left|S_4^{out}\right| + \left|E^{cross}(S_3, S_4)\right|\right) \cdot 4s \\
&+ \left(\left|E(S_1, S_2)\right| + \left|E(S_1, S_4)\right|\right) \cdot (4s - \left|S_3^{out}\right| - \left|S_4^{out}\right|) \\
&+ \left(\left|E(S_2, S_3)\right|\right) \cdot (3s - \left|S_3^{out}\right| - \left|S_4^{out}\right|) \\
&+ \left|E^{in}(S_3, S_4)\right| \cdot (\left|S_3^{in}\right| + \left|S_4^{in}\right|) + \left|E^{out}(S_3, S_4)\right| \cdot (\left|S_3^{out}\right| + \left|S_4^{out}\right|) \\
&+ \frac{2}{3}(s^3 - s) + \frac{1}{3}(\left|S_3^{in}\right|^3 + \left|S_4^{in}\right|^3 + \left|S_3^{out}\right|^3 + \left|S_4^{out}\right|^3 - 2s).
\end{aligned}$$

Note that the above expressions are based on the assumption that $\left|E(S_1, S_4^{in})\right| + \left|E(S_2, S_3^{in})\right| \leq \left|E(S_3^{in}, S_4^{in})\right|$, and we now remove this assumption by using a uniform lower bound. Note that no matter how we switch the order of split, the edges $E(S_1, S_2)$, $E(S_1, S_4)$, and $E(S_2, S_3)$ have to pay a multiplicative factor of at least $2s$. Furthermore, the edges $E^{in}(S_3, S_4)$ and $E^{out}(S_3, S_4)$ have to pay the multiplicative factors of $(\left|S_3^{in}\right| + \left|S_4^{in}\right|)$ and $(\left|S_3^{out}\right| + \left|S_4^{out}\right|)$, respectively. Therefore, we can establish a lower bound for $\mathcal{C}_2$ regardless the order of split:

$$\begin{aligned}
\mathcal{C}_2 > &\left(\left|S_3^{in}\right| \cdot \left|S_3^{out}\right| + \left|S_4^{in}\right| \cdot \left|S_4^{out}\right| + \left|E^{cross}(S_3, S_4)\right|\right) \cdot 4s \\
&+ \left(\left|E(S_1, S_2)\right| + \left|E(S_1, S_4)\right| + \left|E(S_2, S_3)\right|\right) \cdot 2s \\
&+ \left|E^{in}(S_3, S_4)\right| \cdot (\left|S_3^{in}\right| + \left|S_4^{in}\right|) + \left|E^{out}(S_3, S_4)\right| \cdot (\left|S_3^{out}\right| + \left|S_4^{out}\right|)
\end{aligned}$$

$$+ \frac{2}{3}(s^3 - s) + \frac{1}{3}(\left|S_3^{in}\right|^3 + \left|S_4^{in}\right|^3 + \left|S_3^{out}\right|^3 + \left|S_4^{out}\right|^3 - 2s).$$

By merging and canceling out different terms, we can show that

$$\begin{aligned}\mathcal{C}_2 - \mathcal{C}_1 > &\left(\left|S_3^{in}\right| \cdot \left|S_3^{out}\right| + \left|S_4^{in}\right| \cdot \left|S_4^{out}\right|\right) \cdot 4s + \left|E^{cross}(S_3, S_4)\right| \cdot 3s \\ &- (|E(S_1, S_2)| + |E(S_1, S_4)| + |E(S_2, S_3)|) \cdot 2s \\ &- \left|E^{in}(S_3, S_4)\right| \cdot (\left|S_3^{out}\right| + \left|S_4^{out}\right|) - \left|E^{out}(S_3, S_4)\right| \cdot (\left|S_3^{in}\right| + \left|S_4^{in}\right|) \\ &- (\left|S_3^{in}\right| \cdot \left|S_3^{out}\right| + \left|S_4^{in}\right| \cdot \left|S_4^{out}\right|) \cdot s,\end{aligned}$$

where the second line comes from the gap of the costs for the edges $E(S_1, S_2)$, $E(S_1, S_4)$ and $E(S_2, S_3)$, the third line comes from the gap for the $\left|E^{in}(S_3, S_4)\right|$ and $\left|E^{out}(S_3, S_4)\right|$ edges, and the final line comes from the gap between $\frac{2}{3}(s^3 - s)$ and $\frac{1}{3}(\left|S_3^{in}\right|^3 + \left|S_4^{in}\right|^3 + \left|S_3^{out}\right|^3 + \left|S_4^{out}\right|^3 - 2s)$. We can upper bound the absolute value of the third line by

$$\begin{aligned}\left|E^{in}(S_3, S_4)\right| \cdot &(\left|S_3^{out}\right| + \left|S_4^{out}\right|) + \left|E^{out}(S_3, S_4)\right| \cdot (\left|S_3^{in}\right| + \left|S_4^{in}\right|) \\ &\leq \left|S_3^{in}\right|\left|S_4^{in}\right| \cdot (\left|S_3^{out}\right| + \left|S_4^{out}\right|) + \left|S_3^{out}\right|\left|S_4^{out}\right| \cdot (\left|S_3^{in}\right| + \left|S_4^{in}\right|) \\ &\qquad \text{(since } \left|E^{in}(S_3, S_4)\right| \leq \left|S_3^{in}\right|\left|S_4^{in}\right| \text{ and } \left|E^{out}(S_3, S_4)\right| \leq \left|S_3^{out}\right|\left|S_4^{out}\right|\text{)} \\ &= s \cdot \left(\left|S_3^{in}\right| \cdot \left|S_3^{out}\right| + \left|S_4^{in}\right| \cdot \left|S_4^{out}\right|\right).\end{aligned}$$

Therefore, the gap between the costs is at least

$$\begin{aligned}\mathcal{C}_2 - \mathcal{C}_1 &> \left(\left|S_3^{in}\right| \cdot \left|S_3^{out}\right| + \left|S_4^{in}\right| \cdot \left|S_4^{out}\right|\right) \cdot 2s - (|E(S_1, S_2)| + |E(S_1, S_4)| + |E(S_2, S_3)|) \cdot 2s \\ &\geq \left(\left|S_3^{in}\right| \cdot \left|S_3^{out}\right| + \left|S_4^{in}\right| \cdot \left|S_4^{out}\right| - \frac{3}{4} \cdot s\right) \cdot 2s \\ &\qquad\qquad (|E(S_1, S_2)| + |E(S_1, S_4)| + |E(S_2, S_3)| \leq \tfrac{3}{4} \cdot s) \\ &\geq 2s^2 > 0. \qquad \text{(since } \left|S_3^{in}\right| \cdot \left|S_3^{out}\right| \geq s - 1 \text{ and } \left|S_4^{in}\right| \cdot \left|S_4^{out}\right| \geq s - 1\text{)}\end{aligned}$$

That is, the quantity of $\mathcal{C}_2 - \mathcal{C}_1$ is always *positive*. Hence, a tree the pattern of $\mathcal{T}$ is strictly better than following the pattern of $\mathcal{T}'$, which means the optimal tree will not start the first split involving edges inside S_3 and S_4. Hence, the first cut of a optimal tree should be restricted to $E(S_1, S_2) \cup E(S_2, S_3) \cup E(S_1, S_4) \cup E(S_3, S_4)$.

We then prove that, conditioning the first cut splits the edges among $E(S_1, S_2) \cup E(S_2, S_3) \cup E(S_1, S_4) \cup E(S_3, S_4)$, the second cut is also restricted to the edges between cliques. Once again, let $\mathcal{T}$ be the best possible tree that first splits the edges between cliques (and therefore edges among $E(S_2, S_3)$), and let $\mathcal{T}'$ be the best possible tree that first split edges inside S_3 and S_4. By the same induction argument, we show the second cut is not inside S_3 or S_4. Denote $\mathcal{C}_1$ as the cost of $\mathcal{T}$ and $\mathcal{C}_2$ as the cost of $\mathcal{T}'$, we have

$$\begin{aligned}\mathcal{C}_2 - \mathcal{C}_1 &\geq \left(\left|S_3^{in}\right| \cdot \left|S_3^{out}\right| + \left|S_4^{in}\right| \cdot \left|S_4^{out}\right|\right) \cdot 2s - (2s - \left|S_3^{out}\right| - \left|S_4^{out}\right|) \cdot (\left|S_3^{out}\right| \cdot \left|S_4^{out}\right|) \\ &\quad - (2s - \left|S_3^{in}\right| - \left|S_4^{in}\right|) \cdot (\left|S_3^{in}\right| \cdot \left|S_4^{in}\right|) - |E(S_2, S_3)| \cdot 2s \\ &= \left(\left|S_3^{in}\right| \cdot \left|S_3^{out}\right| + \left|S_4^{in}\right| \cdot \left|S_4^{out}\right|\right) \cdot s - |E(S_2, S_3)| \cdot 2s \\ &\geq 2(s-1) \cdot s - \frac{3s}{8} \cdot 2s > 0, \qquad (\left|S_3^{in}\right| \cdot \left|S_3^{out}\right| \geq s - 1, |E(S_2, S_3)| \leq \tfrac{3s}{8})\end{aligned}$$

where the equality is obtained by again using $(\left|S_3^{in}\right| - \left|S_4^{in}\right|) \cdot (\left|S_3^{out}\right| \cdot \left|S_4^{out}\right|) + (\left|S_3^{out}\right| - \left|S_4^{out}\right|) \cdot \left|S_3^{in}\right| \cdot (\left|S_4^{in}\right|) = \left(\left|S_3^{in}\right| \cdot \left|S_3^{out}\right| + \left|S_4^{in}\right| \cdot \left|S_4^{out}\right|\right) \cdot s$. As a result, the second cut among any optimal tree is restricted to the edges among $E(S_1, S_2) \cup E(S_2, S_3) \cup E(S_1, S_4) \cup E(S_3, S_4)$. ■

Proof [Proof of Proposition 65] By Lemmas 69 and 70, we effectively rule out any optimal clustering tree $\mathcal{T}$ that cuts clique edges in its first two splits. Therefore, by Lemma 68, the optimal HC tree first separates S_1 from the rest of the graph and obtains G', and then separate S_2 from the rest of the graph to obtain graph G''. Finally, by Proposition 62, the optimal hierarchical clustering tree on G'' must first separate S_3 and S_4. As such, the behavior of the optimal HC tree is exactly as characterized in Proposition 65. ■

Appendix G. A Weaker Version of Lemma 19

In this section, we present a weaker version of Lemma 19 with a $O(\log n)$ approximation factor. The value of the weaker version is that (i) the proof is much simpler; and (ii) it gives some results on binary tree analysis in addition to the edge cost charging as we used in Lemma 19, which may be of independent interests.

The formal statement of the weaker result is as follows.

Proposition 71 *For any graph $G = (V, E, w)$, there exists a $(2/3)$-balanced tree $\mathcal{T}_{balanced}$ such that*

$$\mathbf{cost}_G(\mathcal{T}_{balanced}) \leq O(\log n) \cdot \mathsf{OPT}(G).$$

We first use the balanced minimum cut problem to lower bound the cost of optimum solution.

Lemma 72 *For any graph G and any tree HC-tree $\mathcal{T}$,*

$$\mathbf{cost}_G(\mathcal{T}) \geq n/3 \cdot \min_{S, \bar{S} \subseteq V} w(S, \bar{S}) \quad \textit{s.t.} \quad n/3 \leq |S|, \left|\bar{S}\right| \leq 2n/3.$$

Proof Since $\mathcal{T}$ is a binary tree with n leaf-nodes, there should exists a node u in $\mathcal{T}$ with

$$n/3 \leq |\textbf{leaf-nodes}(\mathcal{T}[u])| \leq 2n/3;$$

(the proof is a standard vertex separator argument for binary trees). Let us fix that node u and consider the node w as the parent of u in $\mathcal{T}$; let $(A, B) = \textbf{cut}(\mathcal{T}[w])$ with A being the side of cut assigned to u, i.e., $A = \textbf{leaf-nodes}(\mathcal{T}[u])$.

Now consider the cut $(A, \bar{A})$ which is a global cut of G. For any edge (x, y) of this cut, $x \vee y$ is either w or some node on the path from the root to w. This, combined with Eq (1), implies that

$$\mathbf{cost}_G(\mathcal{T}) \geq w(A, \bar{A}) \cdot |A \cup B| \geq w(A, \bar{A}) \cdot n/3,$$

as $|A| \geq n/3$. Moreover, the cut $(A, \bar{A})$ satisfies the property that $n/3 \leq |A|, \left|\bar{A}\right| \leq 2n/3$. As such, the minimum in RHS of the lemma statement is at most $w(A, \bar{A})$, which implies the lemma. ■

Proof [Proof of Prop. 71] Consider the following process for constructing $\mathcal{T}_{balanced}$:

(i) Pick a cut $(S, \bar{S})$ of G minimizing $w(S, \bar{S})$ subject to $n/3 \leq |S|, |\bar{S}| \leq 2n/3$. Let the root r of $\mathcal{T}_{balanced}$ be such that **cut**$(\mathcal{T}_{balanced}[r]) = (S, \bar{S})$ (this uniquely identifies the root).

(ii) Let G_S and $G_{\bar{S}}$ be the induced subgraphs of G on S and $\bar{S}$, respectively. Recursively run the same process for G_S and $G_{\bar{S}}$ and let the root of their corresponding trees be the left-child and right-child node of r, respectively (the base case is when the sets have size 1 in which case they form leaf-nodes of $\mathcal{T}_{balanced}$).

It is clear that $\mathcal{T}_{balanced}$ is valid HC-tree for G and that it is $(2/3)$-balanced by Definition 17, simply by the "splitting rule" of part (i). Moreover, $\mathcal{T}_{balanced}$ being $(2/3)$-balanced implies that the depth of this tree is $O(\log n)$, which we will use in proving the upper bound on the cost of the tree.

Consider all nodes $u_1, \ldots, u_t$ at some depth d of the tree (for some $d = O(\log n)$). For $i \in [t]$, let G_i be the induced subgraph of G on vertices in **leaf-nodes**$(\mathcal{T}[u_i])$ and $(S_i, \bar{S}_i)$ be the cut chosen for this node in the process above. By Observation 11,

$$\sum_{i=1}^{t} \mathsf{OPT}(G_i) \leq \mathsf{OPT}(G).$$

At the same time, by Lemma 72, for every $i \in [t]$,

$$\mathsf{OPT}(G_i) \geq 1/3 \cdot \left|S_i \cup \bar{S}_i\right| \cdot w(S_i, \bar{S}_i),$$

which, together with the previous bound, implies that

$$\sum_{i=1}^{t} \left|S_i \cup \bar{S}_i\right| \cdot w(S_i, \bar{S}_i) \leq 3 \cdot \mathsf{OPT}(G).$$

Finally, by combining this with Eq (2), we have that,

$$\mathbf{cost}_G(\mathcal{T}_{balanced}) = \sum_{d=1}^{O(\log n)} \sum_{i=1}^{t_d} \left|S_i \cup \bar{S}_i\right| \cdot w(S_i, \bar{S}_i) \leq \sum_{d=1}^{O(\log n)} 3 \cdot \mathsf{OPT}(G) = O(\log n) \cdot \mathsf{OPT}(G),$$

as the depth of the tree is $O(\log n)$. This concludes the proof. ■