AIM: An Adaptive and Iterative Mechanism for Differentially Private Synthetic Data

Ryan McKenna
Brett Mullins
University of Massachusetts
Amherst, Massachusetts
{rmckenna,bmullins}@cs.umass.edu

Daniel Sheldon
Gerome Miklau
University of Massachusetts
Amherst, Massachusetts
{sheldon,miklau}@cs.umass.edu

ABSTRACT

We propose AIM, a new algorithm for differentially private synthetic data generation. AIM is a workload-adaptive algorithm within the paradigm of algorithms that first selects a set of queries, then privately measures those queries, and finally generates synthetic data from the noisy measurements. It uses a set of innovative features to iteratively select the most useful measurements, reflecting both their relevance to the workload and their value in approximating the input data. We also provide analytic expressions to bound per-query error with high probability which can be used to construct confidence intervals and inform users about the accuracy of generated data. We show empirically that AIM consistently outperforms a wide variety of existing mechanisms across a variety of experimental settings.

PVLDB Reference Format:
Ryan McKenna, Brett Mullins, Daniel Sheldon, and Gerome Miklau. AIM: An Adaptive and Iterative Mechanism for Differentially Private Synthetic Data. PVLDB, 15(11): 2599 - 2612, 2022.
doi:10.14778/3551793.3551817

1 INTRODUCTION

Differential privacy [14] has grown into the preferred standard for privacy protection, with significant adoption by both commercial and governmental enterprises. Many common computations on data can be performed in a differentially private manner, including aggregates, statistical summaries, and the training of a wide variety predictive models. Yet one of the most appealing uses of differential privacy is the generation of synthetic data, which is a collection of records matching the input schema, intended to be broadly representative of the source data. Differentially private synthetic data is an active area of research [1, 2, 4, 10, 11, 18, 24, 26, 29, 30, 45, 47, 48, 50–52, 54–57] and has also been the basis for two competitions, hosted by the U.S. National Institute of Standards and Technology [43].

Private synthetic data is appealing because it fits any data processing workflow designed for the original data, and, on its face, the user may believe they can perform *any* computation they wish, while still enjoying the benefits of privacy protection. Unfortunately, it is well-known that there are limits to the accuracy that can be provided by synthetic data under differential privacy or any other reasonable notion of privacy [13].

As a consequence, it is important to tailor synthetic data to some class of tasks, and this is commonly done by asking the user to provide a set of queries, called the workload, to which the synthetic data can be tailored. However, as our experiments will show, existing workload-aware techniques often fail to outperform workload-agnostic mechanisms, even when evaluated specifically on their target workloads. Not only do these algorithms fail to produce accurate synthetic data, but they provide no way for end-users to detect the inaccuracy. As a result, in practical terms, differentially private synthetic data generation remains an unsolved problem.

In this work, we advance the state-of-the-art of differentially private synthetic data in two key ways. First, we propose a new workload-aware mechanism that offers lower error than all competing techniques. Second, we derive analytic expressions to bound the per-query error of the mechanism with high probability.

Our mechanism, AIM, follows the select-measure-generate paradigm, which can be used to describe many prior approaches.[1] Mechanisms following this paradigm first *select* a set of queries, then *measure* those queries in a differentially private way (through noise addition), and finally *generate* synthetic data consistent with the noisy measurements. We leverage Private-PGM [40] for the generate step, as it provides a robust and efficient method for combining the noisy measurements into a single consistent representation from which records can be sampled.

The low error of AIM is primarily due to innovations in the *select* stage. AIM uses an iterative, greedy selection procedure, inspired by the popular MWEM algorithm for linear query answering. Through careful analysis, we define a low-sensitivity quality score function to determine the best marginal to measure next, which takes into account: (i) how well the candidate marginal is already estimated, (ii) the expected improvement measuring it can offer, (iii) the relevance of the marginal to the workload, and (iv) the available privacy budget. This new quality score is accompanied by a host of other algorithmic techniques including adaptive selection of rounds and budget-per-round, intelligent initialization, and new set of candidates from which to select.

In conjunction with AIM, we develop new techniques to quantify the uncertainty in query answers derived from the generated synthetic data. The bounds on error are useful in practice to understand which queries the synthetic data supports well, and which it does not, and are therefore critical to avoid the mis-use of the

This work is licensed under the Creative Commons BY-NC-ND 4.0 International License. Visit https://creativecommons.org/licenses/by-nc-nd/4.0/ to view a copy of this license. For any use beyond those covered by this license, obtain permission by emailing info@vldb.org. Copyright is held by the owner/author(s). Publication rights licensed to the VLDB Endowment.
Proceedings of the VLDB Endowment, Vol. 15, No. 11 ISSN 2150-8097.
doi:10.14778/3551793.3551817

[1] Another common approach is based on GANs [19]. Recent research [46] has shown that published GAN-based approaches rarely outperform simple baselines; therefore, we do not compare with those techniques in this paper.

Table 1: Table of Notation

Symbol	Meaning
ϵ, δ, ρ	Privacy parameters
Ω	Domain
d	Number of attributes
x	Single record in Ω
D	Dataset of records in Ω
r	Subset of attributes
M_r	Marginal query
n_r	Domain size of attributes r
W	Workload (marginal queries + weights
c_r	Weight on marginal r in the workload
Δ	Sensitivity
p	Distribution over Ω
$\mathcal{S}$	Set of all distributions over Ω
T	Number of rounds of MWEM+PGM
α	Privacy budget split of AIM
w_r	Weighted assigned to marginal r by AIM

data by downstream users, a danger that could limit the adoption of synthetic data [27]. To the best of our knowledge, AIM is the first synthetic data mechansim equipped with such guarantees.The problem of error quantification for data independent mechanisms like the Laplace or Gaussian mechanism is trivial, as they provide unbiased answers with known variance to all queries. The problem is considerably more challenging for data-dependent mechanisms like AIM, where complex post-processing is performed and only a subset of workload queries have unbiased answers. Some mechanisms, like MWEM, provide theoretical guarantees on their worst-case error, under suitable assumptions. However, this is an *a priori* bound on error obtained from a theoretical analysis of the mechanism under worst-case datasets. Instead, we develop an *a posteriori* error analysis, derived from the intermediate differentially private measurements used to produce the synthetic data. Our error estimates therefore reflect the actual execution of AIM on the input data but do not require any additional privacy budget for their calculation.

This paper makes the following contributions:

(1) In Section 3, we assess the prior work in the field, characterizing different approaches via key distinguishing elements and limitations, which brings clarity to a complex space.
(2) In Section 4, we propose AIM, a new mechanism for synthetic data generation that is workload-aware (for workloads consisting of weighted marginals) as well as data-aware.
(3) In Section 5, we derive analytic expressions to bound the per-query error of AIM with high probability. These expressions can be used to construct confidence bounds.
(4) In Section 6, we conduct a comprehensive empirical evaluation and show that AIM consistently outperforms all prior work, improving error over the next best mechanism by 1.6× on average and up to 5.7× in some cases.

2 BACKGROUND

In this section we provide the requisite background on datasets, marginals, and differential privacy needed to understand this work.

2.1 Data, Marginals, and Workloads

Data. A dataset D is a multiset of N records, each containing potentially sensitive information about one individual. Each record $x \in D$ is a d-tuple $(x_1, \dots, x_d)$. The domain of possible values for x_i is denoted by Ω_i, which we assume is finite and has size $|\Omega_i| = n_i$. The full domain of possible values for x is thus $\Omega = \Omega_1 \times \cdots \times \Omega_d$ which has size $\prod_i n_i = n$. We use $\mathcal{D}$ to denote the set of all possible datasets, which is equal to $\cup_{N=0}^{\infty} \Omega^N$.

Marginals. A marginal is a central statistic to the techniques studied in this paper, as it captures low-dimensional structure common in high-dimensional data distributions. A marginal for a set of attributes r is essentially a histogram over x_r: it is a table that counts the number of occurrences of each $t \in \Omega_r$.

Definition 1 (Marginal)**.** *Let $r \subseteq [d]$ be a subset of attributes, $\Omega_r = \prod_{i \in r} \Omega_i$, $n_r = |\Omega_r|$, and $x_r = (x_i)_{i \in r}$. The marginal on r is a vector $\mu \in \mathbb{R}^{n_r}$, indexed by domain elements $t \in \Omega_r$, such that each entry is a count, i.e., $\mu[t] = \sum_{x \in D} \mathbb{1}[x_r = t]$. We let $M_r : \mathcal{D} \rightarrow \mathbb{R}^{n_r}$ denote the function that computes the marginal on r, i.e., $\mu = M_r(D)$.*

In this paper, we use the term *marginal query* to denote the function M_r, and *marginal* to denote the vector of counts $\mu = M_r(D)$. With some abuse of terminology, we will sometimes refer to the attribute subset r as a marginal query as well.

Workload. A workload is a collection of queries the synthetic data should preserve well. It represents the measure by which we will evaluate utility of different mechanisms. We want our mechanisms to take a workload as input and adapt intelligently to the queries in it, providing synthetic data that is tailored to the queries of interest. In this work, we focus on the special (but common) case where the workload consists of a collection of weighted marginal queries. Our utility measure is stated in Definition 2.

Definition 2 (Workload Error)**.** *A workload W consists of a list of marginal queries $r_1, \dots, r_k$ where $r_i \subseteq [d]$, together with associated weights $c_i \geq 0$. The error of a synthetic dataset $\hat{D}$ is defined as:*

$$Error(D, \hat{D}) = \frac{1}{k \cdot |D|} \sum_{i=1}^{k} c_i \left\| M_{r_i}(D) - M_{r_i}(\hat{D}) \right\|_1$$

We measure error using a normalized L_1 distance between the true workload query answers and the synthetic workload query answers. This L_1 error metric is a common choice [7, 40, 54, 57]; although, alternatives have been considered in prior work including L_∞ error [3, 32, 33, 50] and L_2 (squared) error [11, 36]. The L_1 metric is appealing because it captures the overall error better than the L_∞ metric, and is easily interpretable. We also provide supplemental evaluations with L_∞ and L_2 error in Appendix J of the full paper.

2.2 Differential Privacy

Differential privacy protects individuals by bounding the impact any one individual can have on the output of an algorithm. This is formalized using the notion of neighboring datasets. Two datasets $D, D' \in \mathcal{D}$ are neighbors (denoted $D \sim D'$) if D' can be obtained from D by adding or removing a single record.

Definition 3 (Differential Privacy)**.** *A randomized mechanism $\mathcal{M} : \mathcal{D} \rightarrow \mathcal{R}$ satisfies (ϵ, δ)-differential privacy (DP) if for any* *neighboring datasets $D \sim D' \in \mathcal{D}$, and any subset of possible outputs $S \subseteq \mathcal{R}$,*

$$\Pr[\mathcal{M}(D) \in S] \leq \exp(\epsilon) \Pr[\mathcal{M}(D') \in S] + \delta.$$

A key quantity needed to reason about the privacy of common randomized mechanisms is the *sensitivity*, defined below.

Definition 4 (Sensitivity). *Let $f : \mathcal{D} \to \mathbb{R}^p$ be a vector-valued function of the input data. The L_2 sensitivity of f is $\Delta(f) = \max_{D \sim D'} \|f(D) - f(D')\|_2$.*

It is easy to verify that the L_2 sensitivity of any marginal query M_r is 1, regardless of the attributes in r. This is because one individual can only contribute a count of one to a single cell of the output vector. Below we introduce the two building block mechanisms used in this work.

Definition 5 (Gaussian Mechanism). *Let $f : \mathcal{D} \to \mathbb{R}^p$ be a vector-valued function of the input data. The Gaussian Mechanism adds i.i.d. Gaussian noise with scale $\sigma\Delta(f)$ to each entry of $f(D)$. That is,*

$$\mathcal{M}(D) = f(D) + \sigma\Delta(f)\mathcal{N}(0, \mathbb{I}),$$

where $\mathbb{I}$ is a $p \times p$ identity matrix.

Definition 6 (Exponential Mechanism). *Let $q_r : \mathcal{D} \to \mathbb{R}$ be quality score function defined for all $r \in \mathcal{R}$ and let $\epsilon \geq 0$ be a real number. Then the exponential mechanism outputs a candidate $r \in \mathcal{R}$ according to the following distribution:*

$$\Pr[\mathcal{M}(D) = r] \propto \exp\left(\frac{\epsilon}{2\Delta} \cdot q_r(D)\right),$$

where $\Delta = \max_{r \in \mathcal{R}} \Delta(q_r)$.

Our algorithm is defined using zCDP, an alternate version of differential privacy definition which offers beneficial composition properties. We convert to (ϵ, δ) guarantees when necessary.

Definition 7 (zero-Concentrated Differential Privacy (zCDP)). *A randomized mechanism $\mathcal{M}$ is ρ-zCDP if for any two neighboring datasets D and D', and all $\alpha \in (1, \infty)$, we have:*

$$D_\alpha(\mathcal{M}(D) \,||\, \mathcal{M}(D')) \leq \rho \cdot \alpha,$$

where D_α is the Rényi divergence of order α.

Proposition 1 (zCDP of the Gaussian Mechanism [5]). *The Gaussian Mechanism satisfies $\frac{1}{2\sigma^2}$-zCDP.*

Proposition 2 (zCDP of the Exponential Mechanism [9]). *The Exponential Mechanism satisfies $\frac{\epsilon^2}{8}$-zCDP.*

We rely on the following propositions to reason about multiple adaptive invocations of zCDP mechanisms, and the translation from zCDP to (ϵ, δ)-DP. The proposition below covers 2-fold adaptive composition of zCDP mechanisms, and it can be inductively applied to obtain analogous k-fold adaptive composition guarantees.

Proposition 3 (Adaptive Composition of zCDP Mechanisms [5]). *Let $\mathcal{M}_1 : \mathcal{D} \to \mathcal{R}_1$ be ρ_1-zCDP and $\mathcal{M}_2 : \mathcal{D} \times \mathcal{R}_1 \to \mathcal{R}_2$ be ρ_2-zCDP. Then the mechanism $\mathcal{M} = \mathcal{M}_2(D, \mathcal{M}_1(D))$ is $(\rho_1 + \rho_2)$-zCDP.*

Proposition 4 (zCDP to DP [8]). *If a mechanism $\mathcal{M}$ satisfies ρ-zCDP, it also satisfies (ϵ, δ)-differential privacy for all $\epsilon \geq 0$ and*

$$\delta = \min_{\alpha > 1} \frac{\exp\left((\alpha - 1)(\alpha\rho - \epsilon)\right)}{\alpha - 1} \left(1 - \frac{1}{\alpha}\right)^\alpha.$$

2.3 Private-PGM

An important component of our approach is a tool called Private-PGM [35, 37, 40]. For the purposes of this paper, we will treat Private-PGM as a black box that exposes an interface for solving subproblems important to our mechanism. We briefly summarize Private-PGM and three core utilities it provides. Private-PGM consumes as input a collection of noisy marginals of the sensitive data, in the format of a list of tuples $(\tilde{\mu}_i, \sigma_i, r_i)$ for $i = 1, \dots, k$, where $\tilde{\mu}_i = M_{r_i}(D) + \mathcal{N}(0, \sigma_i^2 \mathbb{I})$.[2]

Distribution Estimation. At the heart of Private-PGM is an optimization problem to find a distribution $\hat{p}$ that "best explains" the noisy observations $\tilde{\mu}_i$:

$$\hat{p} \in \underset{p \in \mathcal{S}}{\operatorname{argmin}} \sum_{i=1}^{k} \frac{1}{\sigma_i} \left\|M_{r_i}(p) - \tilde{\mu}_i\right\|_2^2$$

Here $\mathcal{S} = \{p \mid p(x) \geq 0 \text{ and } \sum_{x \in \Omega} p(x) = n\}$ is the set of (scaled) probability distributions over the domain Ω.[3] When $\tilde{\mu}_i$ are corrupted with i.i.d. Gaussian noise, this is exactly a maximum likelihood estimation problem [35, 37, 40]. In general, convex optimization over the scaled probability simplex is intractable for the high-dimensional domains we are interested in. Private-PGM overcomes this curse of dimensionality by exploiting the fact that the objective only depends on p through its marginals. The key observation is that one of the minimizers of this problem is a graphical model $\hat{p}_\theta$. The parameters θ provide a compact representation of the distribution p that we can optimize efficiently.

Junction Tree Size. The time and space complexity of Private-PGM depends on the measured marginal queries in a nuanced way, the main factor being the size of the junction tree implied by the measured marginal queries [37, 39]. While understanding the junction tree construction is not necessary for this paper, it is important to note that Private-PGM exposes a callable function JT-SIZE$(r_1, \dots, r_k)$ that can be invoked to check how large a junction tree is. JT-SIZE is measured in megabytes, and the runtime of distribution estimation is roughly proportional to this quantity. If arbitrary marginals are measured, JT-SIZE can grow out of control, no longer fitting in memory, and leading to unacceptable runtime.

Synthetic Data Generation. Given an estimated model $\hat{p}$, Private-PGM implements a routine for generating synthetic tabular data that approximately matches the given distribution. It achieves this with a randomized rounding procedure, which is a lower variance alternative to sampling from $\hat{p}$ [37].

3 PRIOR WORK ON SYNTHETIC DATA

In this section we survey the state of the field, describing basic elements of a good synthetic data mechanism, along with novelties of more sophisticated mechanisms. We focus our attention on *marginal-based approaches* to differentially private synthetic data in this section, as these have generally seen the most success in practical applications. These mechanisms include PrivBayes [54], PrivBayes+PGM [40], MWEM+PGM [40], MST [37], PrivSyn [57],

[2]Private-PGM is more general than this, but this is the most common setting.
[3]When using unbounded DP, n is sensitive and therefore we must estimate it.

Algorithm 1 MWEM+PGM

Input: Dataset D, workload W, privacy parameter ρ
Output: Synthetic Dataset $\hat{D}$
Hyper-Parameters: rounds $T = d$, budget split $\alpha = 0.9$
Initialize $\hat{p}_0 = \text{Uniform}[\mathcal{X}]$
$\epsilon = \sqrt{8(1-\alpha)\rho/T}$
$\sigma = \sqrt{T/2\alpha\rho}$
for $t = 1, \dots, T$ **do**
  select $r_t \in W$ using exponential mechanism with ϵ budget:

$$q_r(D) = \|M_r(D) - M_r(\hat{p}_{t-1})\|_1 - n_r$$

  measure marginal on C:

$$\tilde{\mu}_t = M_{r_t}(D) + \mathcal{N}(0, \sigma^2 \mathbb{I})$$

  estimate data distribution using Private-PGM:

$$\hat{p}_t = \underset{p \in S}{\text{argmin}} \sum_{i=1}^{t} \left\|M_{r_i}(p) - y_i\right\|_2^2$$

end for
generate synthetic data $\hat{D}$ using Private-PGM:
return $\hat{D}$

RAP [3], GEM [33], and PrivMRF [7]. We will begin with a formal problem statement:

Problem 1 (Workload Error Minimization)**.** *Given a workload W, our goal is to design an (ϵ, δ)-DP synthetic data mechanism $\mathcal{M} : \mathcal{D} \rightarrow \mathcal{D}$ such that the expected error defined in Definition 2 is minimized.*

3.1 The Select-Measure-Generate Paradigm

We begin by providing a broad overview of the basic approach employed by many differentially private mechanisms for synthetic data. These mechanisms all fit naturally into the *select-measure-generate* framework. This framework represents a class of mechanisms which can naturally be broken up into 3 steps: (1) *select* a set of queries, (2) *measure* those queries using a noise-addition mechanism, and (3) *generate* synthetic data that explains the noisy measurements well. We consider iterative mechanisms that alternate between the select and measure step to be in this class as well. Mechanisms within this class differ in their methodology for selecting queries, the noise mechanism used, and the approach to generating synthetic data from the noisy measurements.

MWEM+PGM, shown in Algorithm 1, is one mechanism from this class that serves as a concrete example as well as the starting point for our improved mechanism, AIM. As the name implies, MWEM+PGM is a scalable instantiation of the well-known MWEM algorithm [21] for linear query answering, where the multiplicative weights (MW) step is replaced by a call to Private-PGM. It is a greedy, iterative mechanism for workload-aware synthetic data generation, and there are several variants. One variant is shown in Algorithm 1. The mechanism begins by initializing an estimate of the joint distribution to be uniform over the data domain. Then, it runs for T rounds, and in each round it does three things: (1) selects (via the exponential mechanism) a marginal query that is poorly approximated under the current estimate, (2) measures the selected marginal using the Gaussian mechanism, and (3) estimates a new data distribution (using Private-PGM) that explains the noisy measurements well. After T rounds, the estimated distribution is used to generate synthetic tabular data. MWEM+PGM represents one mechansim from this broad class, but many others are very closely related to it. In fact, RAP and GEM can both be seen as scalable instantiations of MWEM, that use different algorithms to estimate the data distribution instead of Private-PGM. PrivMRF is also closely related to MWEM+PGM (and uses Private-PGM), with some minor differences in design decisions in other parts of the algorithm. Algorithms like PrivBayes, MST, and PrivSyn are also conceptually similar to MWEM+PGM, as they attempt to select marginal queries that are poorly approximated under a simple model. While all of these algorithms are conceptually similar, each one makes different design decisions that may have important performance implications in practice. In the subsequent subsections, we will characterize existing mechanisms in terms of how they approach these different aspects of the problem, and discuss some of the design decisions made by these mechansism.

3.2 Basic Elements of a Good Mechanism

In this section we outline some basic criteria reasonable mechanisms should satisfy to get good performance. These recommendations primarily apply to the *measure* step.

Measure Entire Marginals. Marginals are an appealing statistic to measure because every individual contributes a count of one to exactly one cell of the marginal. As a result, we can measure every cell of $M_r(D)$ at the same privacy cost of measuring a single cell. With a few exceptions [3, 33, 50], existing mechanisms utilize this property of marginals or can be extended to use it. The alternative of measuring a single counting query at a time sacrifices utility unnecessarily.

Use Gaussian Noise. Back of the envelope calculations reveal that if the number of measurements is greater than roughly $\log(1/\delta) + \epsilon$, which is often the case, then the standard deviation of the required Gaussian noise is lower than that of the Laplace noise. Many newer mechanisms recognize this and use Gaussian noise, while older mechanisms were developed with Laplace noise, but can easily be adapted to use Gaussian noise instead.

Use Unbounded DP. For fixed (ϵ, δ), the required noise magnitude is lower by a factor of $\sqrt{2}$ when using unbounded DP (add / remove one record) over bounded DP (modify one record). This is because the L_2 sensitivity of a marginal query M_r is 1 under unbounded DP, and $\sqrt{2}$ under bounded DP. We remark that these two different definitions of DP are qualitatively different, and because of that, the privacy parameters have different interpretations. The $\sqrt{2}$ difference could be recovered in bounded DP by increasing the privacy budget appropriately. In some cases, the privacy model is imposed externally, in which case it is better if the mechanism naturally supports both bounded and unbounded DP. When either privacy definition is acceptable, as in recent NIST competitions [43], unbounded DP should be preferred.

Table 2: **Taxonomy of select-measure-generate mechanisms.**

Name	Year	Workload Aware	Data Aware	Budget Aware	Efficiency Aware
Independent	-				✓
Gaussian	-	✓			
PrivBayes [54]	2014		✓	✓	✓
HDMM+PGM [40]	2019	✓			
PrivBayes+PGM [40]	2019		✓	✓	✓
MWEM+PGM [40]	2019	✓	✓		
PrivSyn [57]	2020		✓	✓	✓
MST [37]	2021		✓		✓
RAP [3]	2021	✓	✓		✓
GEM [33]	2021	✓	✓		✓
PrivMRF [7]	2021		✓	✓	✓
AIM [This Work]	2022	✓	✓	✓	✓

Devote more Budget to the Measure Step. For mechanisms that select marginal queries based on the data, the privacy budget must be split between the select step and the measure step. A simple 50/50 split is usually suboptimal, and it is often better to allocate the majority of the privacy budget for the measure step. Indeed, prior work has reported 10/90 splits to work well empirically in a variety of settings [7, 57]. Intuitively, this uneven split makes sense because the statistics needed to select marginal queries are often coarser grained aggregations than the marginal queries themselves, and as a result are more robust to noise.

Summary. The implementation of MWEM+PGM in Algorithm 1 gets these basic elements right. This particular implementation of MWEM+PGM is new — the original measured a single counting query per round, used Laplace noise, bounded DP, and an even select/measure budget split [21, 40]. While the modifications made are simple, as we will show in Section 6.3, they have a substantial influence on the performance of the mechanism in practice.

3.3 Distinguishing Elements of Existing Work

Beyond the basics, different mechanisms exhibit different novelties, and understanding the design considerations underlying the existing work can be enlightening. We provide a simple taxonomy of this space in Table 2 in terms of four criteria: workload-, data-, budget-, and efficiency-awareness. These characteristics primarily pertain to the *select* step of each mechanism.

Workload-awareness. Different mechanisms select from a different set of candidate marginal queries. PrivBayes and PrivMRF, for example, select from a particular subset of k-way marginals, determined from the data. Other mechanisms, like MST and PrivSyn, restrict the set of candidates to 2-way marginal queries. On the other end of the spectrum, the candidates considered by MWEM+PGM, RAP, and GEM, are exactly the marginal queries in the workload. This is appealing, since these mechanisms will not waste the privacy budget to measure marginals that are not relevant to the workload.

Data-awareness. Many mechanisms select marginal queries from a set of candidates based on the data, and are thus data-aware. For example, MWEM+PGM selects marginal queries using the exponential mechanism with a quality score function that depends on the data. Independent, Gaussian, and HDMM+PGM are the exceptions, as they always select the same marginal queries no matter what the underlying data distribution is.

Budget-awareness. Another aspect of different mechanisms is how well do they adapt to the privacy budget available. Some mechanisms, like PrivBayes, PrivSyn, and PrivMRF recognize that we can afford to measure more (or larger) marginals when the privacy budget is sufficiently large. When the privacy budget is limited, these mechanisms recognize that fewer (and smaller) marginals should be measured instead. In contrast, the number and size of the marginals selected by mechanisms like MST, MWEM+PGM, RAP, and GEM does not depend on the privacy budget available.[4]

Efficiency-awareness. Mechanisms that build on top of Private-PGM must take care when selecting measurements to ensure JT-SIZE remains sufficiently small to ensure computational tractability. Among these, PrivBayes+PGM, MST, and PrivMRF all have built-in heuristics in the selection criteria to ensure the selected marginal queries give rise to a tractable model. Gaussian, HDMM+PGM and MWEM+PGM have no such safeguards, and they can sometimes select marginal queries that lead to intractable models. In the extreme case, when the workload is all 2-way marginals, Gaussian selects all 2-way marginals, the model required for Private-PGM explodes to the size of the entire domain, which is often intractable.

Mechanisms that utilize different techniques for post-processing noisy marginals into synthetic data, like PrivSyn, RAP, and GEM, do not have this limitation, and are free to select from a wider collection of marginals. While these methods do not suffer from this particular limitation of Private-PGM, they have other pros and cons which were surveyed in a recent article [35].

Summary. With the exception of our new mechanism AIM, no mechanism listed in Table 2 is aware of all four factors we discussed. Mechanisms that do not have four checkmarks in Table 2 are not necessarily bad, but there are clear ways in which they can be improved. Conversely, mechanisms that have more checkmarks than other mechanisms are not necessarily better. For example, RAP has 3 checkmarks, but as we show in Section 6, it does not consistently beat Independent, which only has 1 checkmark.

3.4 Other Design Considerations

Beyond these four characteristics summarized in the previous section, different methods make different design decisions that are relevant to mechanism performance, but do not correspond to the four criteria discussed in the previous section. In this section, we summarize some of those additional design considerations.

Selection method. Some mechanisms select marginals to measure in a *batch*, while other mechanisms select them *iteratively*. Generally speaking, iterative methods like MWEM+PGM, RAP, GEM, and PrivMRF are preferable to batch methods, because the selected marginals will capture important information about the distribution that was not effectively captured by the previously measured marginals. On the other hand, PrivBayes, MST, and PrivSyn select all the marginals before measuring any of them. It is not difficult to construct examples where a batch method like PrivSyn has

[4]The number of rounds to run MWEM+PGM, RAP, and GEM is a hyper-parameter, and the best setting of this hyper-parameter depends on the privacy budget available.

suboptimal behavior. For example, suppose the data contains three perfectly correlated attributes. We can expect iterative methods to capture the distribution after measuring any two 2-way marginals. On the other hand, a batch method like PrivSyn will determine that all three 2-way marginals need to be measured.

Budget split. Every mechanism in this discussion, except for PrivSyn, splits the privacy budget equally among selected marginals. This is a simple and natural thing to do, but it does not account for the fact that larger marginals have smaller counts that are less robust to noise, requiring a larger fraction of the privacy budget to answer accurately. PrivSyn provides a simple formula for dividing privacy budget among marginals of different sizes, but this approach is inherently tied to their batch selection methodology. It is much less clear how to divide the privacy budget within a mechanism that uses an iterative selection procedure.

Hyperparameters. All mechanisms have some hyperparameters than can be tuned to affect the behavior of the mechanism. Mechanisms like PrivBayes, MST, PrivSyn, and PrivMRF have reasonable default values for these hyperparameters, and these mechanisms can be expected to work well out of the box. On the other hand, MWEM+PGM, RAP, and GEM have to tune the number of rounds to run, and it is not obvious how to select this a priori. While the open source implementations may include a default value, the experiments conducted in the respective papers did not use these default values, in favor of non-privately optimizing over this hyperparameter for each dataset and privacy level considered [3, 33].

4 AIM: AN ADAPTIVE AND ITERATIVE MECHANISM FOR SYNTHETIC DATA

While MWEM+PGM is a simple and intuitive algorithm, it leaves significant room for improvement, even after getting the basic elements right. Our new mechanism, AIM, is presented in Algorithm 2. In this section, we describe the differences between MWEM+PGM and AIM, the justifications for the relevant design decisions, as well as prove the privacy of AIM.

Intelligent Initialization. In Line 7 of AIM, we spend a small fraction of the privacy budget to measure 1-way marginals in the set of candidates. Estimating $\hat{p}$ from these noisy marginals gives rise to an *independent* model where all 1-way marginals are preserved well, and higher-order marginals can be estimated under an independence assumption. Intuitively, this feature of AIM is justified by the fact that MWEM+PGM tends to select marginal queries covering disjoint attribute subsets in the first few rounds in an attempt to correctly preserve the 1-way marginal distributions. By measuring all 1-way marginals immediately instead, we are saving the privacy budget that would otherwise be spent to select these marginal queries.

New Candidates. In Line 13 of AIM, we make two notable modifications to the candidate set that serve different purposes. Specifically, the set of candidates is a carefully chosen subset of the marginal queries in the *downward closure* of the workload. The downward closure of the workload is the set of marginal queries whose attribute sets are subsets of some marginal query in the workload, i.e., $W_+ = \{r \mid r \subseteq s, s \in W\}$.

Algorithm 2 AIM: An Adaptive and Iterative Mechanism

1: **Input:** Dataset D, workload W, privacy parameter ρ
2: **Output:** Synthetic Dataset $\hat{D}$
3: **Hyper-Parameters:** MAX-SIZE=80MB, $T = 16d$, $\alpha = 0.9$
4: $\sigma_0 = \sqrt{T/(2\,\alpha\,\rho)}$
5: $\rho_{used} = 0$
6: $t = 0$
7: Initialize $\hat{p}_t$ using Algorithm 3
8: $w_r = \sum_{s \in W} c_s \mid r \cap s \mid$
9: $\sigma_{t+1} \leftarrow \sigma_0 \quad \epsilon_{t+1} \leftarrow \sqrt{8(1-\alpha)\rho/T}$
10: **while** $\rho_{used} < \rho$ **do**
11: $\quad t = t + 1$
12: $\quad \rho_{used} \leftarrow \rho_{used} + \frac{1}{8}\epsilon_t^2 + \frac{1}{2\sigma_t^2}$
13: $\quad C_t = \{r_t \in W_+ \mid \text{JT-SIZE}(r_1, \dots, r_t)) \leq \frac{\rho_{used}}{\rho} \cdot \text{MAX-SIZE}\}$
14: $\quad$ **select** $r_t \in C_t$ using the exponential mechanism with:

$$q_r(D) = w_r\Big(\|M_r(D) - M_r(\hat{p}_{t-1})\|_1 - \sqrt{2/\pi} \cdot \sigma_t \cdot n_r \Big)$$

15: $\quad$ **measure** marginal on r_t:

$$\tilde{y}_t = M_{r_t}(D) + \mathcal{N}(0, \sigma_t^2 \mathbb{I})$$

16: $\quad$ **estimate** data distribution using Private-PGM:

$$\hat{p}_t = \operatorname*{argmin}_{p \in S} \sum_{i=1}^{t} \frac{1}{\sigma_i} \left\| M_{r_i}(p) - \tilde{y}_i \right\|_2^2$$

17: $\quad$ anneal ϵ_{t+1} and σ_{t+1} using Algorithm 4
18: **end while**
19: **generate** synthetic data $\hat{D}$ from $\hat{p}_t$ using Private-PGM
20: **return** $\hat{D}$

Algorithm 3 Initialize p_t (subroutine of Algorithm 2)

for $r \in \{r \in W_+ \mid |r| = 1\}$ **do**
$\quad t = t + 1 \quad \sigma_t \leftarrow \sigma_0 \quad r_t \leftarrow r$
$\quad \tilde{y}_t = M_r(D) + \mathcal{N}(0, \sigma_t^2 \mathbb{I})$
$\quad \rho_{used} \leftarrow \rho_{used} + \frac{1}{2\sigma_t^2}$
end for
$\hat{p}_t = \operatorname{argmin}_{p \in S} \sum_{i=1}^{t} \frac{1}{\sigma_i} \left\| M_{r_i}(p) - \tilde{y}_i \right\|_2^2$

Using the downward closure is based on the observation that marginals with many attributes have low counts, and answering them directly with a noise addition mechanism may not provide an acceptable signal to noise ratio. In these situations, it may be better to answer lower-dimensional marginals, as these tend to exhibit a better signal to noise ratio, while still being useful to estimate the higher-dimensional marginals in the workload.

We filter candidates from this set that do not meet a specific model capacity requirement. Specifically, the set will only consist of candidates that, if selected, will lead to a JT-SIZE below a prespecified limit (the default is 80 MB). This ensures that AIM will never select candidates that lead to an intractable model, and hence allows the mechanism to execute consistently with a predictable memory footprint and runtime.

Algorithm 4 Budget annealing (subroutine of Algorithm 2)

1: **if** $\left\|M_{r_t}(\hat{p}_t) - M_{r_t}(\hat{p}_{t-1})\right\|_1 \leq \sqrt{2/\pi} \cdot \sigma_t \cdot n_{r_t}$ **then**
2: $\quad \epsilon_{t+1} \leftarrow 2 \cdot \epsilon_t$
3: $\quad \sigma_{t+1} \leftarrow \sigma_t / 2$
4: **else**
5: $\quad \epsilon_{t+1} \leftarrow \epsilon_t$
6: $\quad \sigma_{t+1} \leftarrow \sigma_t$
7: **end if**
8: **if** $(\rho - \rho_{used}) \leq 2\left(\frac{1}{2\sigma_{t+1}^2} + \frac{1}{8}\epsilon_{t+1}^2\right)$ **then**
9: $\quad \epsilon_{t+1} = \sqrt{8 \cdot (1-\alpha) \cdot (\rho - \rho_{used})}$
10: $\quad \sigma_{t+1} = \sqrt{1/(2 \cdot \alpha \cdot (\rho - \rho_{used}))}$
11: **end if**

Better Selection Criteria. In Line 14 of AIM, we make two modifications to the quality score function for marginal query selection to better reflect the utility we expect from measuring the selected marginal. In particular, our new quality score function is

$$q_r(D) = w_r\left(\|M_r(D) - M_r(p_{t-1})\|_1 - \sqrt{2/\pi} \cdot \sigma_t \cdot n_r \right), \qquad (1)$$

which differs from MWEM+PGM's quality score function $q_r(D) = \|M_r(D) - M_r(p_{t-1})\| - n_r$ in two ways.

First, the expression inside parentheses can be interpreted as the *expected improvement* in L_1 error we can expect by measuring that marginal. It consists of two terms: the L_1 error under the current model minus the expected L_1 error if it is measured at the current noise level (Theorem 5 in the full paper [38]). Compared to the quality score function in MWEM+PGM, this quality score function penalizes larger marginals to a much more significant degree, since $\sigma_t \gg 1$ in most cases. Moreover, this modification makes the selection criteria "budget-adaptive", since it recognizes that we can afford to measure larger marginals when σ_t is smaller, and we should prefer smaller marginals when σ_t is larger.

Second, we give different marginal queries different weights to capture how relevant they are to the workload. In particular, we weight the quality score function for a marginal query r using the formula $w_r = \sum_{s \in W} c_s \mid r \cap s \mid$, as this captures the degree to which the marginal queries in the workload overlap with r. In general, this weighting scheme places more weight on marginals involving more attributes. Note that now the sensitivity of q_r is w_r rather than 1. Thus, to apply the exponential mechanism to select a candidate, we use $\Delta_t = \max_{r \in C_t} w_r$. A nice property of using w_r as a multiplicative weight is a certain invariance to how the workload is represented: in particular, the behavior of AIM is identical in the two cases where (1) two copies of a marginal query are included in the workload, (2) the marginal query appears once with a weight of two. This is not true of MWEM+PGM, which generally has different behavior based on how the workload is represented.

This quality score function exhibits an interesting trade-off: the penalty term $\sqrt{2/\pi}\sigma_t n_r$ discourages marginals with more cells, while the weight w_r favors marginals with more attributes. However, if the inner expression is negative, then the larger weight will make it more negative, and much less likely to be selected.

Adaptive Rounds and Budget Split. In Lines 12 and 17 of AIM, we introduce logic to modify the per-round privacy budget as execution progresses, and as a result, eliminate the need to provide the number of rounds up front. This makes AIM hyper-parameter free, relieving practitioners from that often overlooked burden.

Specifically, we use a simple annealing procedure (Algorithm 4) that gradually increases the budget per round when an insufficient amount of information is learned at the current per-round budget. The annealing condition is activated if the difference between $M_{r_t}(\hat{p}_t)$ and $M_{r_t}(\hat{p}_{t-1})$ is small, which indicates that not much information was learned in the previous round. If it is satisfied, then ϵ_t for is doubled, while σ_t is cut in half.

This check can pass for two reasons: (1) there were no good candidates (all scores are low in Equation (1)) in which case increasing σ_t will make more candidates good, and (2) there were good candidates, but they were not selected because there was too much noise in the select step, which can be remedied by increasing ϵ_t. The precise annealing threshold used is $\sqrt{2/\pi} \cdot \sigma_t \cdot n_{r_t}$, which is the expected error of the noisy marginal, and an approximation for the expected error of $\hat{p}_t$ on marginal r. When the available privacy budget is small, this condition will be activated more frequently, and as a result, AIM will run for fewer rounds. Conversely, when the available privacy budget is large, AIM will run for many rounds before this condition activates.

As σ_t decreases throughout execution, quality scores generally increase, and it has the effect of "unlocking" new candidates that previously had negative quality scores. We initialize σ_t and ϵ_t conservatively, assuming the mechanism will be run for $T = 16d$ rounds. This is an upper bound on the number of rounds that AIM will run, but in practice the number of rounds will be much less.

Privacy Analysis. The privacy analysis of AIM utilizes the notion of a *privacy filter* [9, 15, 44], and the algorithm runs until the realized privacy budget spent matches the total privacy budget available, ρ. To ensure that the budget is not over-spent, there is a special condition (Line 8 in Algorithm 4) that checks if the remaining budget is insufficient for two rounds at the current ϵ_t and σ_t parameters. If this condition is satisfied, ϵ_t and σ_t are set to use up all of the remaining budget in one final round of execution.

Theorem 1. *For any $T \geq d$, $\alpha \in (0,1)$, $\rho \geq 0$, AIM satisfies ρ-zCDP.*

Proof. There are three steps in AIM that depend on the sensitive data: initialization, selection, and measurement. The initialization step satisfies ρ_0-zCDP for $\rho_0 = |\{r \in W_+ \mid |r| = 1\}|/2\sigma_0^2 \leq d/2\sigma_0^2 = 2\alpha d\rho/2T \leq \rho$. For this step, all we need is that the privacy budget is not over-spent. The remainder of AIM runs until the budget is consumed. Each step of AIM involves one invocation of the exponential mechanism, and one invocation of the Gaussian mechanism. By Propositions 1 to 3, round t of AIM is ρ_t-zCDP for $\rho_t = \frac{1}{8}\epsilon_t^2/8 + 1/2\sigma_t^2$. Note that at round t, $\rho_{used} = \sum_{i=0}^{t} \rho_i$, and by Theorem 3.1 of [15], it suffices to show that ρ_{used} never exceeds ρ. There are two cases to consider: the condition in Line 8 of Algorithm 4 is either true or false. If it is true, then we know after round t that $\rho - \rho_{used} \geq 2\rho_{t+1}$, i.e., the remaining budget is enough to run round $t+1$ without over-spending the budget. If it is false, then we modify ϵ_{t+1} and ρ_{t+1} to exactly use up the remaining budget. Specifically, $\rho_{t+1} = 8(1-\alpha)(\rho - \rho_{used})/8 + 2\alpha(\rho - \rho_{used})/2 = \rho - \rho_{used}$. As a result, when the condition is true, ρ_{used} at time $t+1$ is exactly

ρ, and after that iteration, the main loop of AIM terminates. The remainder of the mechanism does not access the data. □

5 UNCERTAINTY QUANTIFICATION

We now propose a solution to the uncertainty quantification problem for AIM. Our method uses information from *both* the noisy marginals, measured with Gaussian noise, and the marginal queries selected by the exponential mechanism. The method does not require additional privacy budget, as it quantifies uncertainty only by analyzing the private outputs of AIM. We give guarantees for marginals in the (downward closure of the) workload, which is exactly the set of marginals the analyst cares about. Providing guarantees for marginals outside this set is an area for future work.

We break our analysis up into two cases: the "easy" case, where we have access to unbiased answers for a particular marginal, and the "hard" case, where we do not. In both cases, we identify an *estimator* for a marginal whose error we can bound with high probability. Then, we connect the error of this estimator to the error of the synthetic data by invoking the triangle inequality. Proofs of all statements in this section appear in the full paper [38].

The Easy Case: Supported Marginal Queries. A marginal query r is "supported" whenever $r \subseteq r_t$ for some t. In this case, we can readily obtain an unbiased estimate of $M_r(D)$ from y_t, and analytically derive the variance of that estimate. If there are multiple t satisfying the condition above, we have multiple estimates we can use to reduce the variance. We can combine these independent estimates to obtain a *weighted average estimator*:

Theorem 2 (Weighted Average Estimator). *Let $r_1, \dots, r_t$ and $y_1, \dots, y_t$ be as defined in Algorithm 2, and let $R = \{r_1, \dots, r_t\}$. For any $r \in R_+$, there is an (unbiased) estimator $\bar{y}_r = f_r(y_1, \dots, y_t)$ such that:*

$$\bar{y}_r \sim \mathcal{N}(M_r(D), \bar{\sigma}_r^2 \mathbb{I}) \quad \text{where} \quad \bar{\sigma}_r^2 = \Big[\sum_{\substack{i=1 \\ r \subseteq r_i}}^{t} \frac{n_r}{n_{r_i} \sigma_i^2} \Big]^{-1},$$

While this is not the only (or best) estimator to use [12], the simplicity allows us to easily bound its error, as we show in Theorem 3.

Theorem 3 (Confidence Bound). *Let $\bar{y}_r$ be the estimator from Theorem 2. Then, for any $\lambda \geq 0$, with probability at least $1 - \exp(-\lambda^2)$:*

$$\|M_r(D) - \bar{y}_r\|_1 \leq \sqrt{2 \log 2}\, \bar{\sigma}_r n_r + \lambda \bar{\sigma}_r \sqrt{2 n_r}$$

Note that Theorem 3 gives a guarantee on the error of $\bar{y}_r$, but we are ultimately interested in the error of $\hat{D}$. Fortunately, it easy easy to relate the two by using the triangle inequality:

Corollary 1. *Let $\hat{D}$ be any synthetic dataset, and let $\bar{y}_r$ be the estimator from Theorem 2. Then with probability at least $1 - \exp(-\lambda^2)$:*

$$\left\|M_r(D) - M_r(\hat{D})\right\|_1 \leq \left\|M_r(\hat{D}) - \bar{y}_r\right\|_1 + \sqrt{2 \log 2}\, \bar{\sigma}_r n_r + \lambda \bar{\sigma}_r \sqrt{2 n_r}$$

The LHS is what we are interested in bounding, and we can readily compute the RHS from the output of AIM. The RHS is a random quantity that, with the stated probability, upper bounds the error. When we plug in the realized values we get a concrete numerical bound that can be interpreted as a (one-sided) confidence interval. In general, we expect $M_r(\hat{D})$ to be close to $\bar{y}_r$, so the error bound for $\hat{D}$ will not be that much larger than that of $\bar{y}_r$.[5]

The Hard Case: Unsupported Marginal Queries. We now shift our attention to the hard case, providing guarantees about the error of different marginals even for unsupported marginal queries (those not selected during execution of AIM). This problem is significantly more challenging. Our key insight is that marginal queries *not selected* have relatively low error compared to the marginal queries that were selected. We can easily bound the error of selected queries and relate that to non-selected queries by utilizing the guarantees of the exponential mechanism. In Theorem 4 below, we provide expressions that capture the uncertainty of these marginals with respect to $\hat{p}_{t-1}$, the iterates of AIM.

Theorem 4 (Confidence Bound). *Let $\sigma_t, \epsilon_t, r_t, \tilde{y_t}, C_t, \hat{p}_t$ be as defined in Algorithm 2, and let $\Delta_t = \max_{r \in C_t} w_r$. For all $r \in C_t$, with probability at least $1 - e^{-\lambda_1^2/2} - e^{-\lambda_2}$:*

$$\|M_r(D) - M_r(\hat{p}_{t-1})\|_1 \leq w_r^{-1} \big(B_r + \lambda_1 \sigma_t \sqrt{n_{r_t}} + \lambda_2 \frac{2\Delta_t}{\epsilon_t}\big)$$

where B_r is equal to:

$$\underbrace{w_{r_t} \left\|M_{r_t}(\hat{p}_{t-1}) - y_t\right\|_1}_{\text{estimated error on } r_t} + \underbrace{\sqrt{2/\pi}\sigma_t \big(w_r n_r - w_{r_t} n_{r_t}\big)}_{\substack{\text{relationship to} \\ \text{non-selected candidates}}} + \underbrace{\frac{2\Delta_t}{\epsilon_t} \log(|C_t|)}_{\substack{\text{uncertainty from} \\ \text{exponential mech.}}}$$

We can readily compute B_r from the output of AIM, and use it to provide a bound on error in the form of a one-sided confidence interval that captures the true error with high probability. While these error bounds are expressed with respect to $\hat{p}_{t-1}$, they can readily be extended to give a guarantee with respect to $\hat{D}$.

Corollary 2. *Let $\hat{D}$ be any synthetic dataset, and let B_r be as defined in Theorem 4. Then with probability at least $1 - e^{-\lambda_1^2/2} - e^{-\lambda_2}$:*

$$\begin{aligned} &\left\|M_r(D) - M_r(\hat{D})\right\|_1 \\ &\leq \left\|M_r(\hat{D}) - M_r(\hat{p}_{t-1})\right\|_1 + w_r^{-1} \big(B_r + \lambda_1 \sigma_t \sqrt{n_{r_t}} + \lambda_2 \frac{2\Delta_t}{\epsilon_t}\big) \end{aligned}$$

Again, the LHS is what we are interested in bounding, and we can compute the RHS from the output of AIM. We expect $\hat{p}_{t-1}$ to be reasonably close to $\hat{D}$, especially when t is larger, so this bound will often be comparable to the original bound on $\hat{p}_{t-1}$.

Putting it Together. We've provided guarantees for both supported and unsupported marginals. The guarantees for unsupported marginals also apply for supported marginals, although we generally expect them to be looser. In addition, there is one guarantee *for each round of AIM.* It is tempting to use the bound that provides the smallest estimate, although unfortunately doing this invalidates the bound. To ensure a valid bound, we must pick only one round, and that cannot be decided based on the value of the bound. A natural choice is to use only the last round, for three reasons: (1) σ_t is smallest and ϵ_t is largest in that round, (2) the error of $\hat{p}_t$ generally goes down with t, and (3) the distance between $\hat{p}_t$ and

[5] From prior experience, we might expect the error of $\hat{D}$ to be *lower* than the error of $\bar{y}_r$ [40, 41], so we are paying for this difference by increasing the error bound when we might hope to save instead. Unfortunately, this intuition does not lend itself to a clear analysis that provides better guarantees.

$\hat{D}$ should be the smallest in the last round. However, there may be some marginal queries which were not in the candidate set for that round. To bound the error on these marginals, we use the last round where that marginal query was in the candidate set.

6 EXPERIMENTS

In this section we empirically evaluate AIM, comparing it to a collection of state-of-the-art mechanisms and baseline mechanisms for a variety of workloads, datasets, and privacy levels.

6.1 Experimental Setup

Datasets. Our evaluation includes datasets with varying size and dimensionality. We describe our exact pre-processing scheme in the full paper [38], and summarize the pre-processed datasets and their characteristics in the table below.

Table 3: Summary of datasets used in the experiments.

Dataset	Records	Dimensions	Min/Max Domains	Total Domain Size
ADULT [28]	48842	15	2–42	4×10^{16}
SALARY [23]	135727	9	3–501	1×10^{13}
MSNBC [6]	989818	16	18	1×10^{20}
FIRE [43]	305119	15	2–46	4×10^{15}
NLTCS [34]	21574	16	2	7×10^{4}
TITANIC [17]	1304	9	2–91	9×10^{7}

Workloads. We consider 3 workloads for each dataset, ALL-3WAY, TARGET, and SKEWED. Each workload contains a collection of 3-way marginal queries. The ALL-3WAY workload contains queries for *all* 3-way marginals. The TARGET workload contains queries for all 3-way marginals involving some specified *target* attribute. For the ADULT and TITANIC datasets, these are the INCOME>50K attribute and the SURVIVED attribute, as those correspond to the attributes we are trying to predict for those datasets. For the other datasets, the target attribute is chosen uniformly at random. The SKEWED workload contains a collection of 3-way marginal queries *biased* towards certain attributes and attribute combinations. In particular, each attribute is assigned a weight sampled from a squared exponential distribution. 256 triples of attributes are sampled with probability proportional to the product of their weights. This results in workloads where certain attributes appear far more frequently than others, and is intended to capture the situation where analysts focus on a small number of interesting attributes. In Appendix K, we provide results on a fourth workload, ALL-2WAY as well. All randomness in the construction of the workload was done with a fixed random seed, to ensure that the workloads remain the same across executions of different mechanisms and parameter settings.

Mechanisms. We compare against both workload-agnostic and workload-aware mechanisms in this section. The workload-agnostic mechanisms we consider are PrivBayes+PGM, MST, PrivMRF. The workload-aware mechanisms we consider are MWEM+PGM, RAP, GEM, and AIM. We set the hyper-parameters of every mechanism to default values available in their open source implementations. While these default hyper-parameters may be suboptimal, we conducted sensitivity experiments in Appendix I to evaluate the impact of hyper-parameters on the performance of competing mechanisms, and found that the improvement in utility from optimizing hyper-parameters is outweighed by the cost to privacy needed to run an appropriate DP hyper-parameter selection mechanism. We also consider baseline mechanisms: Independent and Gaussian. The former measures all 1-way marginals using the Gaussian mechanism, and generates synthetic data using an independence assumption. The latter answers all queries in the workload using the Gaussian mechanism (using the optimal privacy budget allocation described in [57]). Note that this mechanism *does not* generate synthetic data, only query answers.

Privacy Budgets. We consider a wide range of privacy parameters, varying $\epsilon \in [0.01, 100.0]$ and setting $\delta = 10^{-9}$. The most practical regime is $\epsilon \in [0.1, 10.0]$, but mechanism behavior at the extremes can be enlightening so we include them as well.

Evaluation. For each dataset, workload, and ϵ, we run each mechanism for 5 trials, and measure the workload error from Definition 2. We report the average workload error across the five trials, along with error bars corresponding to the minimum and maximum workload error observed across the five trials. In Appendix J, we evaluate error using L_2 and L_{inf} error metrics as well.

Runtime Environment. We ran most experiments on a single core of a compute cluster with a 4 GB memory limit and a 24 hour time limit.[6] These resources were not sufficient to run PrivMRF or RAP, so we utilized different machines to run those mechanisms. PrivMRF requires a GPU to run, so we used one node a different compute cluster, which has a Nvidia GeForce RTX 2080 Ti GPU. RAP required significant memory resources, so we ran those experiments on a machine with 16 cores and 64 GB of RAM.

6.2 Experimental Results

Experimental results are shown in Figure 1. Results for the TITANIC dataset are omitted due to space. Workload-aware mechanisms are shown by solid lines, while workload-agnostic mechanisms are shown with dotted lines. Some points are missing from the plots, indicating a mechanism failed to complete in under the 24 hour time limit for that experimental setting. From these plots, we make the following observations:

ALL-3WAY Workload.

(1) AIM consistently achieves competitive workload error, across all datasets and privacy regimes considered. On average, across all six datasets and nine privacy parameters, AIM improved over PrivMRF by a factor of 1.3×, MST by a factor of 2.6×, MWEM+PGM by a factor of 1.5×, PrivBayes+PGM by a factor 2.2×, RAP by a factor 5.6×, and GEM by a factor 2.0×. In the most extreme cases, AIM improved over PrivMRF by a factor 3.6×, MST by a factor 118×, MWEM+PGM by a factor 16×, PrivBayes+PGM by a factor 14.7×, RAP by a factor 47.1×, and GEM by a factor 11.7×.

(2) Prior to AIM, PrivMRF was consistently the best performing mechanism, even outperforming all workload-aware mechanisms. The ALL-3WAY workload is one we expect workload agnostic mechanisms like PrivMRF to perform

[6] These experiments usually completed in well under the time limit.

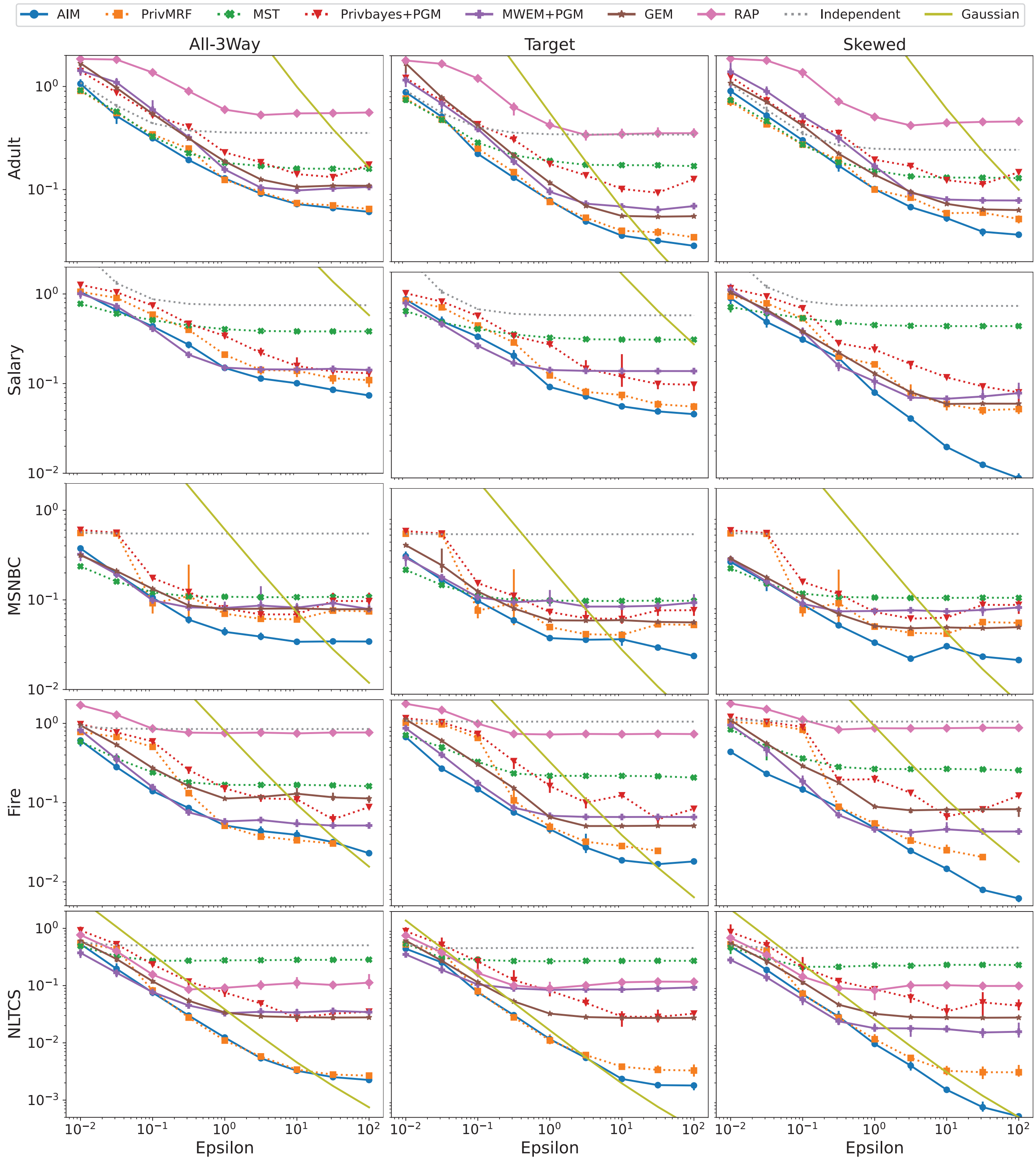


Figure 1: Workload error (y-axis) vs Epsilon (x-axis) of competing mechanisms on the ALL-3WAY (left), TARGET (center), and SKEWED (right) workloads for $\delta = 10^{-9}$.

well on, so it is interesting, but not surprising that it outperforms workload-aware mechanisms in this setting.

(3) Prior to AIM, the best *workload-aware* mechanism varied for different datasets and privacy levels: MWEM+PGM was best in 72% of settings, GEM was best in 28% of settings [7], and RAP was best in 0% of settings. Including AIM, we observe that it is best in 76% of settings, followed by MWEM+PGM in 18% of settings and GEM in 5% of settings. Additionally, in the most interesting regime for practical deployment ($\epsilon \geq 1.0$), AIM is best in 100% of settings.

***target* Workload.**

(1) All three high-level findings from the previous section are supported by these figures as well.
(2) Somewhat surprisingly, PrivMRF outperforms all workload-aware mechanisms prior to AIM on this workload. This is an impressive accomplishment for PrivMRF, and clearly highlights the suboptimality of existing workload-aware mechanisms like MWEM+PGM, GEM, and RAP. Even though PrivMRF is not workload-aware, it is clear from their paper that every detail of the mechanism was carefully thought out to make the mechanism work well in practice, which explains it's impressive performance. While AIM did outperform PrivMRF again, the relative performance did not increase by a meaningful margin — offering a 1.4× improvement on average and a 4.6× improvement in the best case.

***skewed* Workload.**

(1) All four high-level findings from the previous sections are generally supported by these figures as well, with the following interesting exception:
(2) PrivMRF did not score well on salary, and while it was still generally the second best mechanism on the other datasets (again out-performing the workload-aware mechanisms in many cases), the improvement offered by AIM over PrivMRF is much larger for this workload, averaging a 2× improvement with up to a 5.7× improvement in the best case. We suspect for this setting, workload-awareness is essential to achieve strong performance.

6.3 Ablations

In this section, we systematically evaluate the components of AIM, by making modifications to the base mechanism and measuring their impact on workload error. Specifically, the elements we study are enumerated in Figure 2b, and are labeled by B1, B2, and B3 for the basic elements of a good mechanism described in Section 3.2, A1, A2, and A3 for the new elements of AIM described in Section 4, and O1 for an additional relevent element. For each element of AIM listed below, we run AIM with and without that element across the entire set of experimental configurations we considered in this work, i.e., 9 privacy budgets × 6 datasets × 3 workloads × 5 trials. For each of the 162 (privacy budget, dataset, workload) triples, we have 5 measurements which we use to compute two things: (1) the ratio of average workload errors with and without the specified element, and (2) a p-value from a one sided t-test. The former quantity provides a measure of *practical significance*, while the latter quantity provides a measure of *statistical significance*.

Figure 2a shows the distribution of error ratios for each element across experimental settings, visualized as a box-and-whisker plot; ratios above 1 indicate the element of AIM reduced error. Aggregating the error ratios via geometric mean reveals the three basic elements improved error by a factor 1.18 on average for Gaussian noise, 1.13 for unbounded DP, and 1.08 for a 10/90 budget split. The new elements of AIM improved error by a factor of 1.03 for initialization, 1.37 for the new selection critera, and 1.48 for adaptive rounds and budget split. Finally, using Private-PGM in the generate step, rather than an alternative known as relaxed projection [3], improved error by a factor of 2.36 on average. Among these elements, the improvement offered by using adaptive rounds + budget split, as well as Private-PGM, showed a clear depeendence on ϵ, with improvements growing with increasing ϵ. The other elements showed no clear dependence on ϵ.

Aggregating the 162 p-values via Stouffer's Z-score method [53], we see that the combined p-value for every element tested ranges from 10^{-22} for A1 (initialization) all the way to 10^{-166} for A3 (adaptive rounds + budget split). Thus, it is clear that all elements have a positive effect on the performance of AIM in a statistical sense.

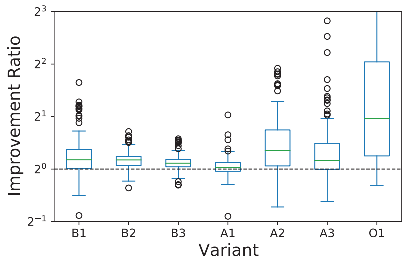
In addition to the algorithmic elements of AIM we evaluate in this section, we conducted experiments varying the model capacity parameter of AIM in Appendix G. Unsurprisingly, the utility of AIM increases with larger model capacities, at the cost of increased runtime.

6.4 Uncertainty Quantification

In this section, we demonstrate that our expressions for uncertainty quantification correctly bound the error, and evaluate how tight the bound is. For this experiment, we ran AIM on the fire dataset with the all-3way workload at $\epsilon = 10$. In Figure 6 (c), we plot the true error of AIM on each marginal in the workload against the error bound predicted by our expressions. We set $\lambda = 1.7$ in Corollary 1, and $\lambda_1 = 2.7$, $\lambda_2 = 3.7$ in Corollary 2, which provides 95% confidence bounds. Our main findings are listed below:

(1) For all marginals in the (downward closure of the) workload, the error bound is always greater than true error. This confirms the validity of the bound, and suggests they are safe to use in practice. Note that even if some errors were above the bounds, that would not be inconsistent with our guarantee, as at a 95% confidence level, the bound could fail to hold 5% of the time. The fact that it doesn't suggests there is some looseness in the bound.
(2) The true errors and the error bounds vary considerably, ranging from 10^{-4} all the way up to and beyond 1. In general, the supported marginals have both lower errors, and lower error bounds than the unsupported marginals, which is not surprising. The error bounds are also *tighter* for the supported marginals. The median ratio between error bound and observed error is 4.4 for supported marginals and 8.3 for unsupported marginals. Intuitively, this makes sense because we know selected marginals should have higher error than non-selected marginals, but the error

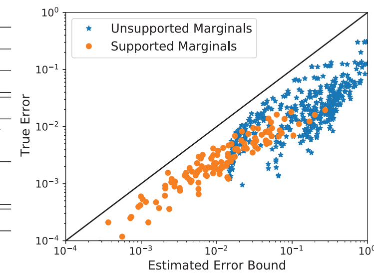
[7]We compare against a variant of GEM that selects an entire marginal query in each round. In results not shown, we also evaluated the variant of that measures a single counting query, and found that this variant performs significantly worse.



(a) Practical Significance

Variant	Element of AIM	Alternative
B1	Gaussian Noise	Laplace Noise
B2	Unbounded DP	Bounded DP
B3	10/90 budget split	50/50 budget split
A1	Independent initialization	Uniform initialization
A2	New selection criteria + candidate set	MWEM+PGM selection + criteria + candidate set
A3	Adaptive rounds + budget split	d rounds + fixed budget per round
O1	Private-PGM	Relaxed Projection

(b) Element codes and descriptions



(c) True Error vs. Error Bound

Figure 2: (a) Box plot of the ratio of errors with and without using an element of AIM across all experimental settings. (b) Table describing elements of AIM removed and the alternatives used in this ablation study. (c) Accuracy of the uncertainty quantification estimates.

of the non-selected marginal can be far below that of the selected marginal (and hence the bound), which explains the larger gap between the actual error and our predicted bound.

7 DISCUSSION AND LIMITATIONS

In this paper, we studied the problem of differentially private synthetic data generation, surveying the field and identifying strengths and weaknesses of prior work. While much of the prior work is conceptually similar, details and specific design decisions differ from mechanism to mechanism, and these small differences can lead to large performance differences in practice. In practical deployments of differential privacy, these details matter to obtain the best privacy-utility trade-off. In this work, we propose AIM, a new mechanism where every detail is carefully thought out to maximize utility in practice. These details allowed AIM to consistently and significantly outperform competitors in our empirical evaluation. In addition, our uncertainty quantification guarantees enables analysts to understand which queries the synthetic data preserves well, and which it does not, which is important to know when performing downstream analyses on synthetic data. While our work significantly improves over prior work, the problem of differentially private synthetic data remains far from solved, and there are a number of promising avenues for future work in this space. We enumerate some of the limitations of AIM below, and identify potential future research directions.

Handling More General Workloads. In this work, we focused on weighted marginal query workloads. Designing mechanisms that work for the more general class of linear queries (perhaps defined over the low-dimensional marginals) remains an important open problem. While the prior work, MWEM+PGM, RAP, and GEM can handle workloads of this form, they achieve this by selecting a single counting query in each round, rather than a full marginal query, and thus there is likely significant room for improvement. Beyond linear query workloads, other workloads of interest include more abstract objectives like machine learning efficacy and other non-linear query workloads. These metrics have been used to evaluate the quality of workload-agnostic synthetic data mechanisms, but have not been provided as input to the mechanisms themselves.

Handling Mixed Data Types. In this work, we assumed the input data was discrete, and each attribute had a finite domain with a reasonably small number of possible values. Data with numerical attributes must be appropriately discretized before running AIM. The quality of the discretization could have a significant impact on the quality of the generated synthetic data. Designing mechanisms that appropriately handle mixed (categorical and numerical) data type is an important problem.

Utilizing Public Data. A promising avenue for future research is to design synthetic data mechanisms that incorporate public data in a principled way. There are many places in which public data can be naturally incorporated into AIM, and exploring these ideas is a promising way to boost the utility of AIM in real world settings where public data is available. Early work on this problem includes [32, 33, 37], but it certainly warrants additional research.

Uncertainty Quantification Guarantees. In this paper, we initiated the study of formal and well-calibrated guarantees about the error of the synthetic data on different marginal queries. These error estimates can be used to determine to what degree the synthetic data should be trusted. However, our guarantees only pertain to the L_1 error of each marginal, and we provide no guarantees on the error in each individual cell of the marginals. These finer-grained guarantees could be useful in some applications, and is an interesting technical challenge for future research.

Small Workloads. In our experimental evaluation, as in much of the current literature, we focus on workloads with a large number of marginal queries where privacy and scalability constraints prevent measuring them all. For smaller workloads, simpler techniques like Gaussian+PGM may achieve better performance than AIM, since it does not have to devote budget to the select step.

High-cardinality attributes. The scalability of AIM, and more generally any method that uses Private-PGM depends on the domain of attributes in the dataset. The datasets we considered in Section 6 were preprocessed to have reasonable domain sizes (most attributes had a domain size ≤ 50). Datasets with high-cardinality attributes often have sparse marginals that may deserve special treatment not covered in this paper.

ACKNOWLEDGMENTS

This work was supported by the National Science Foundation under grants IIS-1749854 and CNS-1954814, and by Oracle Labs, part of Oracle America, through a gift to the University of Massachusetts Amherst in support of academic research.

REFERENCES

[1] Nazmiye Ceren Abay, Yan Zhou, Murat Kantarcioglu, Bhavani M. Thuraisingham, and Latanya Sweeney. 2018. Privacy Preserving Synthetic Data Release Using Deep Learning. In *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2018, Dublin, Ireland, September 10-14, 2018, Proceedings, Part I (Lecture Notes in Computer Science)*, Michele Berlingerio, Francesco Bonchi, Thomas Gärtner, Neil Hurley, and Georgiana Ifrim (Eds.), Vol. 11051. Springer, 510–526. https://doi.org/10.1007/978-3-030-10925-7_31

[2] Hassan Jameel Asghar, Ming Ding, Thierry Rakotoarivelo, Sirine Mrabet, and Mohamed Ali Kâafar. 2019. Differentially Private Release of High-Dimensional Datasets using the Gaussian Copula. *CoRR* abs/1902.01499 (2019). arXiv:1902.01499 http://arxiv.org/abs/1902.01499

[3] Sergul Aydore, William Brown, Michael Kearns, Krishnaram Kenthapadi, Luca Melis, Aaron Roth, and Ankit A Siva. 2021. Differentially Private Query Release Through Adaptive Projection. In *Proceedings of the 38th International Conference on Machine Learning (Proceedings of Machine Learning Research)*, Marina Meila and Tong Zhang (Eds.), Vol. 139. PMLR, 457–467. https://proceedings.mlr.press/v139/aydore21a.html

[4] Vincent Bindschaedler, Reza Shokri, and Carl A. Gunter. 2017. Plausible Deniability for Privacy-Preserving Data Synthesis. *Proceedings of the VLDB Endowment* 10, 5 (2017), 481–492. https://doi.org/10.14778/3055540.3055542

[5] Mark Bun and Thomas Steinke. 2016. Concentrated Differential Privacy: Simplifications, Extensions, and Lower Bounds. In *Theory of Cryptography Conference*. Springer, 635–658. https://doi.org/10.1007/978-3-662-53641-4_24

[6] Igor Cadez, David Heckerman, Christopher Meek, Padhraic Smyth, and Steven White. 2000. Visualization of navigation patterns on a web site using model-based clustering. In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*. 280–284.

[7] Kuntai Cai, Xiaoyu Lei, Jianxin Wei, and Xiaokui Xiao. 2021. Data synthesis via differentially private markov random fields. *Proceedings of the VLDB Endowment* 14, 11 (2021), 2190–2202.

[8] Clément L. Canonne, Gautam Kamath, and Thomas Steinke. 2020. The Discrete Gaussian for Differential Privacy. In *NeurIPS*. https://proceedings.neurips.cc/paper/2020/hash/b53b3a3d6ab90ce0268229151c9bde11-Abstract.html

[9] Mark Cesar and Ryan Rogers. 2021. Bounding, Concentrating, and Truncating: Unifying Privacy Loss Composition for Data Analytics. In *Proceedings of the 32nd International Conference on Algorithmic Learning Theory (Proceedings of Machine Learning Research)*, Vitaly Feldman, Katrina Ligett, and Sivan Sabato (Eds.), Vol. 132. PMLR, 421–457. https://proceedings.mlr.press/v132/cesar21a.html

[10] Anne-Sophie Charest. 2011. How Can We Analyze Differentially-Private Synthetic Datasets? *Journal of Privacy and Confidentiality* 2, 2 (2011). https://doi.org/10.29012/jpc.v2i2.589

[11] Rui Chen, Qian Xiao, Yu Zhang, and Jianliang Xu. 2015. Differentially private high-dimensional data publication via sampling-based inference. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 129–138. https://doi.org/10.1145/2783258.2783379

[12] Bolin Ding, Marianne Winslett, Jiawei Han, and Zhenhui Li. 2011. Differentially private data cubes: optimizing noise sources and consistency. In *Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2011, Athens, Greece, June 12-16, 2011*, Timos K. Sellis, Renée J. Miller, Anastasios Kementsietsidis, and Yannis Velegrakis (Eds.). ACM, 217–228. https://doi.org/10.1145/1989323.1989347

[13] Irit Dinur and Kobbi Nissim. 2003. Revealing information while preserving privacy. In *Proceedings of the Twenty-Second ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, June 9-12, 2003, San Diego, CA, USA*, Frank Neven, Catriel Beeri, and Tova Milo (Eds.). ACM, 202–210. https://doi.org/10.1145/773153.773173

[14] Cynthia Dwork, Frank McSherry Kobbi Nissim, and Adam Smith. 2006. Calibrating Noise to Sensitivity in Private Data Analysis. In *TCC*. 265–284. https://doi.org/10.29012/jpc.v7i3.405

[15] Vitaly Feldman and Tijana Zrnic. 2021. Individual privacy accounting via a renyi filter. *Advances in Neural Information Processing Systems* 34 (2021), 28080–28091.

[16] James S Frame. 1945. Mean deviation of the binomial distribution. *The American Mathematical Monthly* 52, 7 (1945), 377–379.

[17] Thomas Cason Frank E. Harrell Jr. [n.d.]. Encyclopedia Titanica.

[18] Chang Ge, Shubhankar Mohapatra, Xi He, and Ihab F. Ilyas. 2021. Kamino: Constraint-Aware Differentially Private Data Synthesis. *Proceedings of the VLDB Endowment* 14, 10 (2021), 1886–1899. http://www.vldb.org/pvldb/vol14/p1886-ge.pdf

[19] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. 2014. Generative Adversarial Nets. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, Zoubin Ghahramani, Max Welling, Corinna Cortes, Neil D. Lawrence, and Kilian Q. Weinberger (Eds.). 2672–2680. https://proceedings.neurips.cc/paper/2014/hash/5ca3e9b122f61f8f06494c97b1afccf3-Abstract.html

[20] William H Greene. 2003. *Econometric analysis*. Pearson Education India.

[21] Moritz Hardt, Katrina Ligett, and Frank McSherry. 2012. A Simple and Practical Algorithm for Differentially Private Data Release. In *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States*, Peter L. Bartlett, Fernando C. N. Pereira, Christopher J. C. Burges, Léon Bottou, and Kilian Q. Weinberger (Eds.). 2348–2356. https://proceedings.neurips.cc/paper/2012/hash/208e43f0e45c4c78cafadb83d2888cb6-Abstract.html

[22] Joachim Hartung, Guido Knapp, Bimal K Sinha, and Bimal K Sinha. 2008. *Statistical meta-analysis with applications*. Vol. 6. Wiley Online Library.

[23] Michael Hay, Ashwin Machanavajjhala, Gerome Miklau, Yan Chen, and Dan Zhang. 2016. Principled evaluation of differentially private algorithms using dpbench. In *Proceedings of the 2016 International Conference on Management of Data*. 139–154.

[24] Zhiqi Huang, Ryan McKenna, George Bissias, Gerome Miklau, Michael Hay, and Ashwin Machanavajjhala. 2019. PSynDB: accurate and accessible private data generation. *VLDB Demo* (2019). https://people.cs.umass.edu/~miklau/assets/pubs/dp/huang19psyndata-demo.pdf

[25] Norman L Johnson, Adrienne W Kemp, and Samuel Kotz. 2005. *Univariate discrete distributions*. Vol. 444. John Wiley & Sons.

[26] James Jordon, Jinsung Yoon, and Mihaela van der Schaar. 2019. PATE-GAN: Generating Synthetic Data with Differential Privacy Guarantees. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net. https://openreview.net/forum?id=S1zk9iRqF7

[27] Gary King. [n.d.]. Noisy Data from the Noisy Census. ([n. d.]).

[28] Ron Kohavi et al. 1996. Scaling up the accuracy of naive-bayes classifiers: A decision-tree hybrid.. In *Kdd*, Vol. 96. 202–207.

[29] Haoran Li, Li Xiong, and Xiaoqian Jiang. 2014. Differentially Private Synthesization of Multi-Dimensional Data using Copula Functions. In *Proceedings of the 17th International Conference on Extending Database Technology, EDBT 2014, Athens, Greece, March 24-28, 2014*, Sihem Amer-Yahia, Vassilis Christophides, Anastasios Kementsietsidis, Minos N. Garofalakis, Stratos Idreos, and Vincent Leroy (Eds.). OpenProceedings.org, 475–486. https://doi.org/10.5441/002/edbt.2014.43

[30] Fang Liu. 2016. Model-based differentially private data synthesis. *arXiv preprint arXiv:1606.08052* (2016). https://arxiv.org/abs/1606.08052

[31] Jingcheng Liu and Kunal Talwar. 2019. Private selection from private candidates. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*. 298–309.

[32] Terrance Liu, Giuseppe Vietri, Thomas Steinke, Jonathan R. Ullman, and Zhiwei Steven Wu. 2021. Leveraging Public Data for Practical Private Query Release. In *ICML*. 6968–6977. http://proceedings.mlr.press/v139/liu21w.html

[33] Terrance Liu, Giuseppe Vietri, and Steven Wu. 2021. Iterative Methods for Private Synthetic Data: Unifying Framework and New Methods. In *Advances in Neural Information Processing Systems*, A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan (Eds.).

[34] Kenneth G. Manton. 2010. National Long-Term Care Survey: 1982, 1984, 1989, 1994, 1999, and 2004.

[35] Ryan McKenna and Terrance Liu. 2022. A simple recipe for private synthetic data generation. DifferentialPrivacy.org

[36] Ryan McKenna, Gerome Miklau, Michael Hay, and Ashwin Machanavajjhala. 2018. Optimizing error of high-dimensional statistical queries under differential privacy. *Proceedings of the VLDB Endowment* 11, 10 (2018), 1206–1219. https://doi.org/10.14778/3231751.3231769

[37] Ryan McKenna, Gerome Miklau, and Daniel Sheldon. 2021. Winning the NIST Contest: A scalable and general approach to differentially private synthetic data. *Journal of Privacy and Confidentiality* 11, 3 (2021).

[38] Ryan McKenna, Brett Mullins, Daniel Sheldon, and Gerome Miklau. 2022. AIM: An Adaptive and Iterative Mechanism for Differentially Private Synthetic Data. (2022).

[39] Ryan McKenna, Siddhant Pradhan, Daniel R Sheldon, and Gerome Miklau. 2021. Relaxed Marginal Consistency for Differentially Private Query Answering. *Advances in Neural Information Processing Systems* 34 (2021).

[40] Ryan McKenna, Daniel Sheldon, and Gerome Miklau. 2019. Graphical-model based estimation and inference for differential privacy. In *International Conference on Machine Learning*. 4435–4444. http://proceedings.mlr.press/v97/mckenna19a.html

[41] Aleksandar Nikolov, Kunal Talwar, and Li Zhang. 2013. The geometry of differential privacy: the sparse and approximate cases. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*. 351–360.

[42] Nicolas Papernot and Thomas Steinke. 2021. Hyperparameter Tuning with Renyi Differential Privacy. *arXiv preprint arXiv:2110.03620* (2021).

[43] Diane Ridgeway, Mary Theofanos, Terese Manley, Christine Task, et al. 2021. Challenge Design and Lessons Learned from the 2018 Differential Privacy Challenges. (2021).

[44] Ryan M Rogers, Aaron Roth, Jonathan Ullman, and Salil Vadhan. 2016. Privacy odometers and filters: Pay-as-you-go composition. *Advances in Neural Information Processing Systems* 29 (2016), 1921–1929.

[45] Uthaipon Tantipongpipat, Chris Waites, Digvijay Boob, Amaresh Ankit Siva, and Rachel Cummings. 2019. Differentially Private Mixed-Type Data Generation For Unsupervised Learning. *CoRR* abs/1912.03250 (2019). arXiv:1912.03250 http://arxiv.org/abs/1912.03250

[46] Yuchao Tao, Ryan McKenna, Michael Hay, Ashwin Machanavajjhala, and Gerome Miklau. 2021. Benchmarking Differentially Private Synthetic Data Generation Algorithms. *Third AAAI Privacy-Preserving Artificial Intelligence (PPAI-22) workshop* (2021).

[47] Amirsina Torfi, Edward A Fox, and Chandan K Reddy. 2022. Differentially private synthetic medical data generation using convolutional gans. *Information Sciences* 586 (2022), 485–500.

[48] Reihaneh Torkzadehmahani, Peter Kairouz, and Benedict Paten. 2019. DP-CGAN: Differentially Private Synthetic Data and Label Generation. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2019, Long Beach, CA, USA, June 16-20, 2019*. Computer Vision Foundation / IEEE, 98–104. https://doi.org/10.1109/CVPRW.2019.00018

[49] Michail Tsagris, Christina Beneki, and Hossein Hassani. 2014. On the folded normal distribution. *Mathematics* 2, 1 (2014), 12–28.

[50] Giuseppe Vietri, Grace Tian, Mark Bun, Thomas Steinke, and Zhiwei Steven Wu. 2020. New Oracle-Efficient Algorithms for Private Synthetic Data Release. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event (Proceedings of Machine Learning Research)*, Vol. 119. PMLR, 9765–9774. http://proceedings.mlr.press/v119/vietri20b.html

[51] Liyang Xie, Kaixiang Lin, Shu Wang, Fei Wang, and Jiayu Zhou. 2018. Differentially Private Generative Adversarial Network. *CoRR* abs/1802.06739 (2018). arXiv:1802.06739 http://arxiv.org/abs/1802.06739

[52] Chugui Xu, Ju Ren, Yaoxue Zhang, Zhan Qin, and Kui Ren. 2017. DPPro: Differentially Private High-Dimensional Data Release via Random Projection. *IEEE Transactions on Information Forensics and Security* 12, 12 (2017), 3081–3093. https://doi.org/10.1109/TIFS.2017.2737966

[53] Dmitri V Zaykin. 2011. Optimally weighted Z-test is a powerful method for combining probabilities in meta-analysis. *Journal of evolutionary biology* 24, 8 (2011), 1836–1841.

[54] Jun Zhang, Graham Cormode, Cecilia M. Procopiuc, Divesh Srivastava, and Xiaokui Xiao. 2017. PrivBayes: Private Data Release via Bayesian Networks. *ACM Transactions on Database Systems (TODS)* 42, 4 (2017), 25:1–25:41. https://doi.org/10.1145/3134428

[55] Wei Zhang, Jingwen Zhao, Fengqiong Wei, and Yunfang Chen. 2019. Differentially Private High-Dimensional Data Publication via Markov Network. *EAI Endorsed Trans. Security Safety* 6, 19 (2019), e4. https://doi.org/10.4108/eai.29-7-2019.159626

[56] Xinyang Zhang, Shouling Ji, and Ting Wang. 2018. Differentially private releasing via deep generative model (technical report). *arXiv preprint arXiv:1801.01594* (2018). https://arxiv.org/abs/1801.01594

[57] Zhikun Zhang, Tianhao Wang, Ninghui Li, Jean Honorio, Michael Backes, Shibo He, Jiming Chen, and Yang Zhang. 2021. PrivSyn: Differentially Private Data Synthesis. In *30th USENIX Security Symposium (USENIX Security 21)*. USENIX Association, 929–946. https://www.usenix.org/conference/usenixsecurity21/presentation/zhang-zhikun