UNIVERSITY OF CALIFORNIA,
IRVINE

Differentially Private Synthetic Data

DISSERTATION

submitted in partial satisfaction of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

in Mathematics

by

Yiyun He

Dissertation Committee:
Professor Roman Vershynin, Chair
Associate Professor Paata Ivanisvili
Assistant Professor Anna Ma

2025

TABLE OF CONTENTS

LIST OF ALGORITHMS

ACKNOWLEDGMENTS

First of all, I would like to thank my advisor, Professor Roman Vershynin, who has helped me a lot in my Ph.D. life. His book, *High-Dimensional Probability*, introduced me the foundation of high-dimensional probability and provided lots of intuitions at the beginning of my pursuing the doctorate degree. The experiences of reading the book with him have become a solid backing for my current and future research. Professor Vershynin is also the one who introduced the area of differential privacy and various other applications of probability to me. In research, his unique vision of viewing problems is always insightful, showing me the ongoing directions when I get lost in the calculations. Besides research, Professor Vershynin is also very supportive. Financially, he generously funded me so that I can have less teaching load and attend various conferences. Personally, he encouraged me to trust myself in the academic career and held several group celebration events, which have become my valuable memories. During the years working with him, he has passed me his infinite passion in mathematics, and I hope that I can carry this passion and devote myself in mathematical research in the future.

Also, I would like to thank Professor Yizhe Zhu from USC. Professor Zhu used to be a visiting assistant professor in UC, Irvine, and we had two and half years of overlaps. We had several collaborating projects and I really enjoyed the time doing research together. His serious attitude to academics always impressed and influenced me: he is very rigorous and well-considered when writing the paper, and he managed to hold fantastic probability seminars every week without any blank slot. Professor Zhu also helped me a lot in my Ph.D. life. He has recommended many interesting conferences and meetings, and there were multiple times he personally gave me a ride to the conferences nearby. But most importantly, Professor Zhu influenced me to choose the academic career. During the time we spent together, he told me tons of necessary information about being in academics and lots of his insights in mathematical research. Himself sets a perfect goal for me to strive for.

I would like to thank everyone in my family, especially my parents. They are really supportive in my study from my early age till now. They created a cherishable childhood and also later a nice study environment for me. For every accomplishment I have achieved or I would achieve in the future, I will always be grateful for their efforts. Seeing their joy and proud of me makes me full of energy. I wish them staying healthy and happy and I wish I can spend more time with them in the future.

I would also like to thank my girlfriend, Yiming Wang. It is not easy for her to suffer from the two-body problem, but she keeps believing in me and supporting me in the past few years. When I was feeling lazy and wasting the time, she always encouraged me and urged me to try harder. She is also considerate and thoughtful. Even if overwhelmed from work, she was still optimistic in life and passing me positive emotional support. Although I am studying abroad alone, thinking of her keeps me away from loneliness and makes me feel beloved and believe in future. I love her and I feel lucky to be with her.

I would like to thank every friend I met in UC, Irvine, Qizhi Zhao, Jinghao Chen, Doris Yan,

Jongwon Kim, So Nakamura, Yang Yang, and many others. The time spending with them makes my Ph.D. life colorful and memorable. The dinners we enjoyed outside, the board games and video games we played, the special hotpot feasts on the weekend, or simply a random day talking and laughing on random things, I will cherish every moment of it.

Last but least, I would like to thank Professor Thomas Strohmer, Professor Roman Vershynin and Professor Yizhe zhu. Most of the work in the thesis are joint work with them. I would also like to thank the support from ML Research Press, Institute of Mathematics and its Applications in copyright. Also special thanks to [80] providing the template and instructions of arranging the thesis.

VITA

Yiyun He

EDUCATION

Candidate – Doctor of Philosophy in Mathematics **2025**
University of California *Irvine, California, US*

Bachelor of Science in Mathematics **2020**
University of Science and Technology of China *Hefei, Anhui, China*

RESEARCH EXPERIENCE

Graduate Student Researcher **2022–2025**
University of California, Irvine *Irvine, California*

TEACHING EXPERIENCE

Teaching Assistant **2021, 2024**
University of California *Irvine, California, US*

Teaching Assistant **2019**
University of Science and Technology of China *Hefei, Anhui, China*

REFEREED JOURNAL PUBLICATIONS

Differentially private low-dimensional representation of high-dimensional data **2025**
Information and Inferences

Online Differentially Private Synthetic Data Generation **2024**
IEEE Transactions on Privacy

REFEREED CONFERENCE PUBLICATIONS

Algorithmically Effective Differentially Private Synthetic Data **Jul 2023**
Conference on Learning Theory

ABSTRACT OF THE DISSERTATION

Differentially Private Synthetic Data

By

Yiyun He

Doctor of Philosophy in Mathematics

University of California, Irvine, 2025

Professor Roman Vershynin, Chair

Differentially private synthetic data provide a powerful mechanism to enable data analysis while protecting sensitive information about individuals. In this thesis, we present a highly effective algorithmic approach for generating ε-differentially private synthetic data in a bounded metric space with near-optimal utility guarantees under the 1-Wasserstein distance. In particular, for a dataset $\mathcal{X}$ in the hypercube $[0,1]^d$, our algorithm generates synthetic dataset $\mathcal{Y}$ such that the expected 1-Wasserstein distance between the empirical measure of $\mathcal{X}$ and $\mathcal{Y}$ is $O((\varepsilon n)^{-1/d})$ for $d \geq 2$, and is $O(\log^2(\varepsilon n)(\varepsilon n)^{-1})$ for $d = 1$. The accuracy guarantee is optimal up to a constant factor for $d \geq 2$, and up to a logarithmic factor for $d = 1$. Our algorithm has a fast running time of $O(\varepsilon dn)$ for all $d \geq 1$ and demonstrates improved accuracy compared to the method in [14] for $d \geq 2$.

However, when the data lie in a high-dimensional space, the accuracy of the synthetic data suffers from the curse of dimensionality. We further propose a differentially private algorithm to generate low-dimensional synthetic data efficiently from a high-dimensional dataset with a utility guarantee with respect to the 1-Wasserstein distance. A key step of our algorithm is a private principal component analysis (PCA) procedure with a near-optimal accuracy bound that circumvents the curse of dimensionality. Unlike the standard perturbation analysis, our analysis of private PCA works without assuming the spectral gap for the covariance matrix.

For the data lying on a d'-dimensional linear subspace, we successfully overcome the curse of high dimensionality and improve the accuracy to $O(n^{-1/d'})$.

We also consider the synthetic data generation with differential privacy under the online setting where daa is continually released. For a data stream within the hypercube $[0,1]^d$ and an infinite time horizon, we develop an online algorithm that generates a differentially private synthetic dataset at each time t. This algorithm achieves a near-optimal accuracy bound of $O(\log(t)t^{-1/d})$ for $d \geq 2$ and $O(\log^{4.5}(t)t^{-1})$ for $d = 1$ in the 1-Wasserstein distance. This result extends the previous work on the continual release model for counting queries to Lipschitz queries. Compared to the offline case, where the entire dataset is available at once, our approach requires only an extra polynomially logarithmic factor in the accuracy bound.

Chapter 1

Introduction

As data sharing is increasingly locking horns with data privacy concerns, privacy-preserving data analysis is becoming a challenging task with far-reaching impact. Differential privacy (DP) has emerged as the gold standard for implementing privacy in various applications [34]. For instance, DP has been adopted by several technology companies [41] and has also been used in connection with the release of Census 2020 data [2]. The motivation behind the concept of differential privacy is the desire to protect an individual's data while publishing aggregate information about the database.

Private synthetic data Most existing work considered generating differentially private synthetic datasets while minimizing the utility loss for specific queries, including counting queries [11, 52, 36], k-way marginal queries [89, 40], histogram release [1]. For a finite collection of predefined linear queries Q, [52] provided an algorithm with running time linear in $|Q|$ and utility loss grows logarithmically in $|Q|$. The sample complexity can be reduced if the queries are sparse [40, 11, 33]. Beyond finite collections of queries, [96] considered utility bound for differentiable queries, and [14] studied Lipschitz queries with utility bound in Wasserstein distance. [33] considered sparse Lipschitz queries with an improved accuracy

rate. [7, 49, 68, 98] measure the utility of DP synthetic data by the maximum mean discrepancy (MMD) between empirical distributions of the original and synthetic datasets. This metric is different from our chosen utility bound in Wasserstein distance. Crucially, MMD does not provide any guarantees for Lipschitz downstream tasks.

In this work, we focus on the 1-Wasserstein distance to measure the accuracy of the synthetic data. Mathematically, the problem of generating private synthetic data can be defined as follows. Let (Ω, ρ) be a metric space. Consider a dataset $\mathcal{X} = (X_1, \ldots, X_n) \in \Omega^n$. Our goal is to construct an efficient randomized algorithm that outputs differentially private synthetic data $\mathcal{Y} = (Y_1, \ldots, Y_m) \in \Omega^m$ such that the two empirical measures

$$\mu_{\mathcal{X}} = \frac{1}{n}\sum_{i=1}^{n} \delta_{X_i} \quad \text{and} \quad \mu_{\mathcal{Y}} = \frac{1}{m}\sum_{i=1}^{m} \delta_{Y_i}$$

are close to each other. We measure the utility of the output by $\mathbb{E}\, W_1(\mu_{\mathcal{X}}, \mu_{\mathcal{Y}})$, where $W_1(\mu_{\mathcal{X}}, \mu_{\mathcal{Y}})$ is the 1-Wasserstein distance, and the expectation is taken over the randomness of the algorithm. The Kantorovich-Rubinstein duality (see, e.g., [94]) gives an equivalent representation of the 1-Wasserstein distance between two measures $\nu_{\mathcal{X}}$ and $\mu_{\mathcal{Y}}$:

$$W_1(\mu_{\mathcal{X}}, \mu_{\mathcal{Y}}) = \sup_{\mathrm{Lip}(f)\leq 1} \left(\int f \mathrm{d}\mu_{\mathcal{X}} - \int f \mathrm{d}\mu_{\mathcal{Y}} \right), \tag{1.1}$$

where the supremum is taken over the set of all 1-Lipschitz functions on Ω. A more detailed setting can be found in Chapter 2. Given that numerous machine learning algorithms are Lipschitz [95, 67, 18, 77], (1.1) provides data analysts with a vastly increased toolbox of machine learning methods for which one can expect similar outcomes for the original and synthetic data.

1.1 Synthetic data algorithms

[89] proved that it is NP-hard to generate private synthetic data on the Boolean cube which approximately preserves all two-dimensional marginals, assuming the existence of one-way functions. Nonetheless, there exists a substantial body of work for differentially private synthetic data with guarantees limited to accuracy bounds for a finite set of specified queries [8, 87, 40, 90, 71, 93, 14, 15, 16].

[96] considered differentially private synthetic data in $[0,1]^d$ with guarantees for any smooth queries with bounded partial derivatives of order K, and achieved an accuracy bound of $O(\varepsilon^{-1}n^{-\frac{K}{2d+K}})$. [14] introduced a method based on superregular random walks to generate differentially private synthetic data with near-optimal guarantees in general compact metric spaces. In particular, when the dataset is in $[0,1]^d$, they obtain

$$\mathbb{E}\, W_1(\mu_{\mathcal{X}}, \mu_{\mathcal{Y}}) \leq C \log^{\frac{3d}{2}}(\varepsilon n)(\varepsilon n)^{-\frac{1}{d}}.$$

A corresponding lower bound of order $n^{-1/d}$ was also proved in [14, Corollary 9.3]. The algorithm we propose, *Private Measure Mechanism* in Chapter 3, successfully improves the former accuracy and attains the optimal 1-Wasserstein error up to a constant. Moreover, the algorithm has time complexity linear in the size of the input dataset.

The most straightforward way to construct differentially private synthetic data is to add independent noise to the location of each data point. However, this method can result in a significant loss of data utility as the amount of noise needed for privacy protection may be too large [31]. Another direct approach could be to add noise to the density function of the empirical measure of $\mathcal{X}$, by dividing Ω into small subregions and perturbing the true counts in each subregion. However, Laplacian noise may perturb the count in a certain subregion to negative, causing the output to become a signed measure. To address this issue, we introduce

Private Measure Mechanism.

Private Measure Mechanism (PMM) PMM makes the count zero if the noisy count in a subregion is negative. Instead of a single partition of Ω, we consider a collection of binary hierarchical partitions on Ω and add inhomogeneous noise to each level of the partition. However, the counts of two subregions do not always add up to the count of the region at a higher level. We develop an algorithm that enforces the consistency of counts in regions at different levels. PMM has $O(\varepsilon dn)$ running time while the running time of the approach in [14] is polynomial in n.

The accuracy analysis of PMM uses the hierarchical partitions to estimate the 1-Wasserstein distance in terms of the multi-scale geometry of Ω and the noise magnitude in each level of the partition. In particular, when $\Omega = [0,1]^d$, by optimizing the choice of the hierarchical partitions and noise magnitude, PMM achieves better accuracy compared to [14] for $d \geq 2$. The accuracy is optimal rate up to a constant factor for $d \geq 2$, and up to a logarithmic factor for $d = 1$. We state it in the next theorem.

The hierarchical partitions appeared in many previous works on the approximation of distributions under Wasserstein distances in a non-private setting, including [6, 29, 97]. In the differential privacy literature, the hierarchical partitions are also closely related to the binary tree mechanism [37, 22] for differential privacy under continual observation. However, the accuracy analysis of the two mechanisms is significantly different. In addition, the Top-Down algorithm in the 2020 census [2] also has a similar hierarchical structure and enforces consistency, but the accuracy analysis of the algorithm is not provided in [2].

Theorem 1.1 (PMM for data in a hypercube)**.** *Let $\Omega = [0,1]^d$ equipped with the ℓ^∞ metric.*

PMM outputs an ε-differentially private synthetic dataset $\mathcal{Y}$ in time $O(\varepsilon dn)$ such that

$$\mathbb{E}\, W_1(\mu_{\mathcal{X}}, \mu_{\mathcal{Y}}) \leq \begin{cases} C \log^2(\varepsilon n)(\varepsilon n)^{-1} & \text{if } d = 1, \\ C(\varepsilon n)^{-\frac{1}{d}} & \text{if } d \geq 2. \end{cases}$$

Private Signed Measure Mechanism (PSMM) In addition to PMM, we introduce an alternative method, the *Private Signed Measure Mechanism*, that achieves optimal accuracy rate on $[0,1]^d$ when $d \geq 3$ in poly(n) time. The analysis of PSMM is not restricted to 1-Wasserstein distance, and it can be generalized to provide a uniform utility guarantee of other function classes.

We first partition the domain Ω into m subregions $\Omega_1, \ldots, \Omega_m$. Perturbing the counts in each subregion with i.i.d. integer Laplacian noise gives an unbiased approximation of $\mu_{\mathcal{Y}}$ with a signed measure ν. Then we find the closest probability measure $\hat{\nu}$ under the bounded Lipschitz distance by solving a linear programming problem.

In the proof of accuracy for PSMM, one ingredient is to estimate the Laplacian complexity of the Lipschitz function class on Ω and connect it to the 1-Wasserstein distance. This type of argument is similar in spirit to the optimal matching problem for two sets of random points in a metric space [85, 86, 12]. When $\Omega = [0,1]^d$, PSMM achieves the optimal accuracy rate $O((\varepsilon n)^{-1/d})$ for $d \geq 3$. For $d = 2$, PSMM achieves a near-optimal accuracy $O(\log(\varepsilon n)(\varepsilon n)^{-1/2})$. For $d = 1$, the accuracy becomes $O((\varepsilon n)^{-1/2})$.

Note that for the case when $d = 2$, we believe that the bound in Corollary 3.7 could be improved to $C\sqrt{\log(\varepsilon n)}/\sqrt{\varepsilon n}$ by replacing Dudley's chaining bound in Proposition 3.2 with the generic chaining bound in [86, 30] involving the γ_1 and γ_2 functionals on Ω. We will not pursue this direction in this work.

1.2 Synthetic data with low-dimensional input

The result we present in Theorem 1.1 is optimal up to a constant, matching the lower bound in [14]. Even though, such utility guarantee is only useful when d, the dimension of the data, is small (or if the size of dataset n is exponentially larger than d). In other words, we are facing the curse of dimensionality. The curse of dimensionality extends beyond challenges associated with Wasserstein distance utility guarantees. Even with a weaker accuracy requirement, the hardness result from Ullman and Vadhan [89] shows that $n = \text{poly}(d)$ is necessary for generating DP-synthetic data in polynomial time while maintaining approximate covariance.

In [33], the authors succeeded in constructing DP synthetic data with utility bounds where d in Theorem 1.1 is replaced by $(d' + 1)$, assuming that the dataset lies in a certain d'-dimensional subspace. Their notion of dimension is similar to the Minkowski dimension, and their method is applicable beyond the linear subspace setting. However, the optimization step in their algorithm exhibits exponential time complexity in d, see [33, Section D].

This paper presents a computationally efficient algorithm that does not rely on any assumptions about the true data. We demonstrate that our approach enhances the utility bound from d to d' compared to Theorem 1.1 when the dataset is in a d'-dimensional affine subspace. Specifically, we derive a differentially private algorithm to generate low-dimensional synthetic data from a high-dimensional dataset with a utility guarantee with respect to the 1-Wasserstein distance that captures the intrinsic dimension of the data.

Our approach revolves around a private principal component analysis (PCA) procedure with a near-optimal accuracy bound that circumvents the curse of dimensionality. Different from classical perturbation analysis [24, 38] that utilizes the Davis-Kahan theorem [28] in the literature, our accuracy analysis of private PCA works without assuming the spectral gap for the covariance matrix.

Private PCA Private PCA is a commonly used technique for differentially private dimension reduction of the original dataset. This is achieved by introducing noise to the covariance matrix [75, 24, 56, 38, 61, 62, 100]. Instead of independent noise, the method of exponential mechanism is also extensively explored [65, 24, 61]. Another approach, known as streaming PCA [79, 59], can also be performed privately [50, 73].

The private PCA typically yields a private d'-dimensional subspace $\widehat{\mathbf{V}}_{d'}$ that approximates the top d'-dimensional subspace $\mathbf{V}_{d'}$ produced by the standard PCA. The accuracy of private PCA is usually measured by the distance between $\widehat{\mathbf{V}}_{d'}$ and $\mathbf{V}_{d'}$ [38, 51, 75, 73, 83]. To prove a utility guarantee, a common tool is the Davis-Kahan Theorem [10, 99], which assumes that the covariance matrix has a spectral gap [24, 38, 50, 61, 73]. Alternatively, using the projection error to evaluate accuracy is independent of the spectral gap [65, 74, 5]. In our implementation of private PCA, we don't treat $\widehat{\mathbf{V}}_{d'}$ as our terminal output. Instead, we project data matrix $\mathbf{X}$ onto $\widehat{\mathbf{V}}_{d'}$. Our approach directly bound the Wasserstein distance between the projected dataset and $\mathbf{X}$. This method circumvents the subspace perturbation analysis, resulting in an accuracy bound independent of the spectral gap, as outlined in Lemma 4.3. [83] considered a related task that takes a true dataset close to a low-dimensional linear subspace and outputs a private linear subspace. To the best of our knowledge, none of the previous work on private PCA considered low-dimensional DP synthetic data generation.

Centered covariance matrix A common choice of the covariance matrix for PCA is $\frac{1}{n}\mathbf{X}\mathbf{X}^\top$ [23, 38, 83], which is different from the centered one defined in (4.1). The rank of $\mathbf{X}$ is the dimension of the linear subspace that the data lie in rather than that of the affine subspace. If $\mathbf{X}$ lies in a d'-dimensional affine space (not necessarily passing through the origin), centering the data shifts the affine hyperplane spanned $\mathbf{X}$ to pass through the origin. Consequently, the centered covariance matrix will have rank d', whereas the rank of $\mathbf{X}$ is $d'+1$. By reducing the dimension of the linear subspace by 1, the centering step enhances

the accuracy rate from $(\varepsilon n)^{-1/(d'+1)}$ to $(\varepsilon n)^{-1/d'}$. Yet, this process introduces the challenge of protecting the privacy of mean vectors, as detailed in the third step in Algorithm 5 and Algorithm 7.

Private covariance estimation Private covariance estimation [32, 75] is closely linked to the private covariance matrix and the private linear projection components of our Algorithm 5. Instead of adding i.i.d. noise, [65, 4] improved the dependence on d in the estimation error by sampling top eigenvectors with the exponential mechanism. However, it requires d' as an input parameter (in our approach, it can be chosen privately) and a lower bound on $\sigma_{d'}(\mathbf{M})$. The dependence on d is a critical aspect in private mean estimation [64, 72], and it is an open question to determine the optimal dependence on d for low-dimensional synthetic data generation.

1.3 Online data set

Despite extensive research in differential privacy, most advancements have focused on scenarios involving a single collection or release of data. However, in reality, datasets frequently accumulate over time, arriving in a continuous stream rather than being available all at once. This is common in various domains, such as tracking COVID-19 statistics, collecting location data from vehicles [66], or internet search and click data[22]. In these contexts, generating online synthetic data that adheres to differential privacy standards poses significant challenges [20, 69].

One popular model in online differential privacy is the *Continual Release Model*, first studied in [37, 22]. In this model, data points arrive in a streaming fashion, and an online algorithm releases the statistics of the streaming dataset in a differentially private manner. The initial example explored in [37, 22] was for Boolean data streams. At time t, a Boolean sequence

$X_1, \ldots, X_t \in \{0, 1\}$ is available, and private algorithms were developed to release the count $\sum_{i=1}^{t} X_i$ for each $t \leq T$, where T is the time horizon of the streaming data sequence. The scenario when $T = \infty$ is termed the *infinite time horizon*, where the input data stream is an infinite sequence, and the private algorithm outputs an infinite sequence.

In the online setting, repeating offline differentially private counting algorithms would require an increasing privacy budget over time due to the composition property of differential privacy [34], thus not being feasible. A seminal contribution of [37, 22] is the Binary Mechanism, which achieves polylog(t) error while maintaining ε-differential privacy for a finite time horizon T. Additionally, [39] improved the accuracy of the Binary Mechanism to $O\left(\log(T) + \log^{1.5}(n)\right)$ when the Boolean data stream is sparse, i.e., the number of 1's, denoted by n, is much smaller than T. The dependence on T in [39] is optimal and matches the $\Omega(\log T)$ lower bound in [37] for online DP-count release.

The Binary Mechanism serves as a foundational element for many online private optimization problems [48, 63]. Various methods to enhance the Binary Mechanism in different settings have been studied in [22, 39, 81, 44, 54, 55]. Besides counting tasks, DP algorithms for online data have also been discussed for mean estimation [46], moment statistics [78, 42], graph statistics [84], online convex programming [58], decaying sums [17], user stream processing [25], and histograms [21]. Utilizing offline DP algorithms as black boxes, [26] provided a general technique to adapt them to online DP algorithms with utility guarantees.

Our work generalizes the framework of DP-counting queries for Boolean data in the continual release model [37, 22] to online synthetic data generation in a metric space. One of the subroutines, *Inhomogeneous Sparse Counting* (Algorithm 11), is a generalization of the sparse counting mechanism from [39] with inhomogeneous noise according to the online hierarchical partition structure we introduce. The Binary Mechanism in [37, 39] is designed only for a finite time horizon, and a modified Hybrid Mechanism was developed for the infinite case in [22]. Our Algorithm 12 works for data streams with an infinite time horizon.

In terms of online DP synthetic data generation, [69] considered online DP-synthetic data for spatial datasets, and [20] studied online DP synthetic Boolean data with prefixed counting queries under a different notion of differential privacy called zero-concentrated differential privacy (zCDP). Moreover, [20] consider specific types of counting queries and the output dataset at time t is obtained from adding one new data point to the output dataset at time $t-1$. This is different from our synthetic data, where we generate different datasets at different time stamps; see Chapter 2 for more details. To the best of our knowledge, our work is the first to generate online DP-synthetic data with utility guarantees for all Lipschitz queries.

Finally, we discuss the difference between online DP algorithms and their offline counterparts. [19] established a separation for the number of offline, online, and adaptive queries subject to differential privacy. For the continual release model, [60] showed that for certain tasks beyond counting, the accuracy gap between the continual release (online) model and the batch (offline) model is $\tilde{\Omega}(T^{1/3})$, which is much better than the $\Omega(\log T)$ gap shown in [37] between online and offline counting tasks.

Our Theorem 5.1 shows that for a dataset in $[0,1]^d$, the accuracy gap between online and offline DP-synthetic data generation is at most a factor of $O(\log(t))$ for $d \geq 2$, and at most $O(\text{polylog}(t))$ for $d = 1$. The lower bound in [37] also implies an $\Omega(\log(T))/T$ accuracy lower bound in our setting for online DP synthetic data in $[0,1]$ with time horizon T. However, for $d \geq 2$, the argument in [37] cannot be directly generalized to prove an accuracy lower bound for datasets in $[0,1]^d$. We conjecture that when $d \geq 2$, the upper bound in Theorem 5.1 is tight in terms of the dependence on t.

1.4 Organization

In Chapter 2, we introduce the detailed background setting of differential privacy, synthetic data, and related concepts and tools that are frequently applied in the main part. We also present the notation through the thesis. In Chapter 3 we present and discuss our synthetic data algorithm, *Private Measure Mechanism* and *Private Signed Measure Mechanism*. We first consider the general case of generating synthetic data set in a general metric space, and then focus on the special case of dataset from $[0,1]^d$, implying Theorem 1.1. In Chapter 4, we introduce the private PCA algorithm, overcome the curse of high-dimensionality in private synthetic data and improve the accuracy bound with the intrinsic dimension of the dataset. In Chapter 5, we consider the continual release model where the input data is a series depending on the time. We propose the online private synthetic data algorithm and prove that there is only a logarithmic factor in the 1-Wasserstein distance accuracy bound compared to Theorem 1.1. In both Chapter 4 and Chapter 5, we only focus on the special case where the data is from hypercube $[0,1]^d$ for simplicity.

Chapter 2

Preliminaries

2.1 Differential Privacy

The motivation of differential privacy is to apply randomized algorithms such that similar datasets would also induce similar output distributions. We use the following definitions of neighboring datasets and differential privacy from [34].

Definition 2.1 (Neighboring datasets)**.** *Two sets of data $\mathcal{X}$ and $\mathcal{X}'$ are neighbors if $\mathcal{X}, \mathcal{X}'$ differ by at most one element.*

Definition 2.2 (Differential Privacy)**.** *A randomized algorithm $\mathcal{A}$ is ε-differentially private if for any two neighboring datasets $\mathcal{X}, \mathcal{X}'$ and any measurable subset $S \subseteq \mathbb{R}$,*

$$\mathbb{P}\left(\mathcal{A}(\mathcal{X}) \in S\right) \leq \exp(\varepsilon) \cdot \mathbb{P}\left(\mathcal{A}(\mathcal{X}') \in S\right).$$

Here the probability is taken from the probability space of the randomness of $\mathcal{A}$.

2.1.1 Differential privacy under composition

For multiple differentially private algorithms, differential privacy has a useful property that their sequential composition is also differentially private [34, Theorem 3.16].

Lemma 2.3 (Theorem 3.16 in [34]). *Suppose $\mathcal{A}_i$ is ε_i-differentially private for $i = 1, \ldots, m$, then the sequential composition $x \mapsto (\mathcal{A}_1(x), \ldots, \mathcal{A}_m(x))$ is $\sum_{i=1}^m \varepsilon_i$-differentially private.*

Moreover, the following result about *adaptive composition* indicates that algorithms in a sequential composition can use the outputs in the previous steps:

Lemma 2.4 (Theorem 1 in [35]). *Suppose a randomized algorithm $\mathcal{A}_1(x) : \Omega^n \to \mathcal{R}_1$ is ε_1-differentially private, and $\mathcal{A}_2(x, y) : \Omega^n \times \mathcal{R}_1 \to \mathcal{R}_2$ is ε_2-differentially private with respect to the first component for any fixed y. Then the sequential composition*

$$x \mapsto (\mathcal{A}_1(x), \mathcal{A}_2(x, \mathcal{A}_1(x)))$$

is $(\varepsilon_1 + \varepsilon_2)$-differentially private.

2.1.2 Laplacian mechanism

One of the easiest way of ensuring differential privacy is to add Laplacian noises of a certain amount. More precisely, we have the following lemma:

Lemma 2.5 (Inhomogeneous Laplace mechanism). *Let $F : \Omega^n \to \mathbb{R}^k$ be any map, $s = (s_i)_{i=1}^k \in \mathbb{R}_+^k$ be a fixed vector, and $\lambda = (\lambda_i)_{i=1}^k$ be a random vector with independent coordinates $\lambda_i \sim \mathrm{Lap}(s_i)$. Then the map $x \mapsto F(x) + \lambda$ is ε-differentially private, where*

$$\varepsilon = \sup_{x, \tilde{x}} \big\| F(x) - F(\tilde{x}) \big\|_{\ell^1(s)} .$$

Here the supremum is over all pairs of input vectors in Ω^n that differ in one coordinate, and $\|z\|_{\ell^1(s)} = \sum_{i=1}^k |z_i| / s_i$.

Proof. Suppose $x, \tilde{x} \in \Omega^n$ differs in exactly one coordinate. Consider the density functions of the inputs having the same output $y = F(x) + \lambda = F(\tilde{x}) + \tilde{\lambda} \in \mathbb{R}^k$. We have

$$\begin{aligned}
\frac{\mathbb{P}\left\{F(x) + \lambda = y\right\}}{\mathbb{P}\left\{F(\tilde{x}) + \tilde{\lambda} = y\right\}} &= \frac{\mathbb{P}\left\{\lambda = y - F(x)\right\}}{\mathbb{P}\left\{\tilde{\lambda} = y - F(\tilde{x})\right\}} \\
&= \frac{\prod_{i=1}^k \exp\left(-\frac{|(y-F(x))_i|}{s_i}\right)}{\prod_{i=1}^k \exp\left(-\frac{|(y-F(\tilde{x}))_i|}{s_i}\right)} \\
&= \exp\left(-\sum_{i=1}^k \frac{1}{s_i}\left(|(y - F(x))_i| - |(y - F(\tilde{x}))_i|\right)\right) \\
&\leq \exp\left(\|F(x) - F(\tilde{x})\|_{\ell^1(s)}\right) \\
&\leq e^{\varepsilon}
\end{aligned}$$

Therefore, we know $x \mapsto F(x) + \lambda$ is ε-differentially private. □

2.2 Wasserstein distance

Definition 2.6. *For two probability measures μ, ν in a metric space (Ω, ρ) and $p > 1$, the p-Wasserstein distance between them is*

$$W_p(\mu, \nu) \coloneqq \inf_{\gamma \in \Gamma(\mu,\nu)} \left(\int_{\Omega \times \Omega} \rho^p(x, y) \mathrm{d}\gamma(x, y)\right)^{1/p},$$

where $\Gamma(\mu, \nu)$ is the set of all couplings of μ and ν (i.e. γ is a positive measure on Ω^2 and the marginal distributions of γ against the two coordinates are μ, ν respectively).

Another way of interpreting the 1-Wasserstein distance is though the duality. For two prob-

ability measures μ and ν on Ω, we have the *Kantorovich-Rubinstein duality* that gives: (see e.g., [94] for more details)

$$W_1(\mu,\nu) = \sup_{\mathrm{Lip}(f)\leq 1} \Big(\int_\Omega f \mathrm{d}\mu - \int_\Omega f \mathrm{d}\nu \Big), \tag{2.1}$$

where $\mathrm{Lip}(f)$ is the Lipschitz constant of f defined in Section 2.3.

2.3 Lipschitz functions and bounded Lipschitz distance

A function $f : \Omega \to \mathbb{R}$ on a metric space (Ω, ρ) is Lipschitz if there is a constant $L > 0$ such that

$$|f(x) - f(y)| \leq L \cdot \rho(x,y) \quad \forall x, y \in \Omega.$$

The *Lipschitz constant* of f, denoted as $\mathrm{Lip}(f)$, is the infimum of all $L > 0$ such that the inequality holds. Note that $\mathrm{Lip}(\cdot)$ is not a norm as $\mathrm{Lip}(f) = 0$ for any constant valued functions f.

We will next define the Lipschitz norm and the bounded Lipschitz distance. Let the metric space (Ω, ρ) be bounded. The *Lipschitz norm* of a function f is defined as

$$\|f\|_{\mathrm{Lip}} \coloneqq \max\left\{ \mathrm{Lip}(f),\ \frac{\|f\|_\infty}{\mathrm{diam}(\Omega)} \right\}.$$

Let $\mathcal{F}$ be the set of all Lipschitz functions f on Ω with $\|f\|_{\mathrm{Lip}} \leq 1$. For signed measures μ, ν on Ω, we define the *bounded Lipschitz distance*:

$$d_{\mathrm{BL}}(\mu,\nu) \coloneqq \sup_{f\in\mathcal{F}} \left(\int f \mathrm{d}\mu - \int f \mathrm{d}\nu \right).$$

One can easily check that $\|\cdot\|_{\mathrm{Lip}}$ is indeed a norm and d_{BL} is a metric over all signed measures on Ω. Moreover, in the special case where μ and ν are both probability measures, shifting f by a constant does not change the result of $\int f\mathrm{d}\mu - \int f\mathrm{d}\nu$. Therefore, for a bounded domain Ω, we can always assume $f(x_0) = 0$ for a fixed $x_0 \in \Omega$, then $\|f\|_\infty \leq \mathrm{diam}(\Omega)$ when computing the supremum in (2.1). This implies d_{BL}-metric is equivalent to the classical W_1-metric when μ, ν are both probability measures on a bounded domain Ω:

$$W_1(\mu,\nu) = \sup_{\mathrm{Lip}(f)\leq 1}\left(\int f\mathrm{d}\mu - \int f\mathrm{d}\nu\right) = \sup_{\|f\|_{\mathrm{Lip}}\leq 1}\left(\int f\mathrm{d}\mu - \int f\mathrm{d}\nu\right) = d_{\mathrm{BL}}(\mu,\nu). \tag{2.2}$$

2.4 Synthetic data generation

2.4.1 Accuracy of synthetic data

With given dataset $\mathcal{X} = \{X_1, \ldots, X_n\}$ as input, we aim to construct synthetic data algorithms $\mathcal{A}$ with output $\mathcal{A}(\mathcal{X}) = \mathcal{Y} = \{Y_1, \ldots, Y_m\}$ as the synthetic data generated. Here n and m are not necessarily to be distinct. To qualify the accuracy of the algorithms, we consider the 1-Wasserstein distance between the empirical measures of true input dataset and the synthetic output dataset. More precisely, we define

$$\mu_{\mathcal{X}} = \frac{1}{n}\sum_{i=1}^{n}\delta_{X_i}, \quad \mu_{\mathcal{Y}} = \frac{1}{m}\sum_{j=1}^{m}\delta_{Y_j},$$

where δ_x denote the Dirac mass function at point x. Then our goal is to construct synthetic data algorithms $\mathcal{A}$ such that $W_1(\mu_{\mathcal{X}}, \mu_{\mathcal{Y}})$ is bounded. According to the Kantorovich-Rubinstein duality (2.1), we will consider

$$W_1(\mu_{\mathcal{X}}, \mu_{\mathcal{Y}}) = \sup_{\mathrm{Lip}(f)\leq 1}\Big(\int_\Omega f\mathrm{d}\mu_{\mathcal{X}} - \int_\Omega f\mathrm{d}\mu_{\mathcal{Y}}\Big) = \sup_{\mathrm{Lip}(f)\leq 1}\Big(\frac{1}{n}\sum_{i=1}^{n} f(X_i) - \frac{1}{m}\sum_{j=1}^{m} f(Y_j)\Big).$$

2.4.2 Online synthetic data algorithm

Compared to the setting above where the input dataset is acquired at the beginning, we also study a particular setting where new data keeps being updated by time, which is referred as *online data*. In our setting, a dataset is further an infinite sequence

$$\mathcal{X} = (X_1, \dots, X_t, \dots),$$

where each X_t arrives at time $t \in \mathbb{Z}_+$. For simplicity, we will assume that the data X_t is from the hypercube $[0,1]^d$. By Definition 2.1, two data sequences $\mathcal{X}, \mathcal{X}'$ are *neighbors* if they differ in one coordinate. Define the time-t data stream from $\mathcal{X}$ as

$$\mathcal{X}_t = (X_1, \dots, X_t).$$

For each time $t \in \mathbb{Z}_+$, a randomized synthetic data generation algorithm $\mathcal{A}_t$ takes an input $\mathcal{X}_t$ and outputs a synthetic dataset of size t given by

$$\mathcal{Y}_t = (Y_{1,t}, \dots, Y_{t,t}).$$

We note that it is not necessary to keep $\mathcal{Y}_{t-1} \subset \mathcal{Y}_t$; they could be completely disjoint.

Under the online setting, an *online synthetic data generation algorithm* $\mathcal{M}$ with infinite time horizon takes an infinite sequence $\mathcal{X}$ and output an infinite sequence of synthetic datasets such that

$$\mathcal{M}(\mathcal{X}) := (\mathcal{A}_1(\mathcal{X}_1), \dots, \mathcal{A}_t(\mathcal{X}_t), \dots) = (\mathcal{Y}_1, \dots, \mathcal{Y}_t, \dots).$$

We say $\mathcal{M}$ is ε-differentially private if $\mathcal{M}$ satisfies Definition 2.2, which guarantees that the entire sequence of outputs is insensitive to the change of any individual's contribution.

2.5 Binary hierarchical partition

Definition 2.7. *A* binary hierarchical partition *of a set Ω of depth r is a family of subsets Ω_θ indexed by $\theta \in \{0,1\}^{\le r}$, where*

$$\{0,1\}^{\le k} = \{0,1\}^0 \sqcup \{0,1\}^1 \sqcup \cdots \sqcup \{0,1\}^k, \quad k = 0,1,2\ldots,$$

and such that Ω_θ is partitioned into $\Omega_{\theta 0}$ and $\Omega_{\theta 1}$ for every $\theta \in \{0,1\}^{\le r-1}$. By convention, the cube $\{0,1\}^0$ corresponds to $\emptyset$ and we write $\Omega_\emptyset = \Omega$.

When $\theta \in \{0,1\}^j$, we call j the *level* of θ. We can also encode a binary hierarchical partition of Ω in a binary tree of depth r, where the root is labeled Ω and the j-th level of the tree encodes the subsets Ω_θ for θ at level j.

In particular, when $\Omega = [0,1]^d$ equipped with the ℓ_∞-norm and we always partition every subregion into halves by one certain coordinate, the subregion Ω_θ with $\theta \in \{0,1\}^j$ has a volume of 2^{-j} and $\operatorname{diam}(\Omega_\theta) \asymp 2^{-\lfloor j/d \rfloor}$.

2.6 Integer Laplacian distribution

Since our method involves protecting privacy of integer-valued variables, we will use integer Laplacian noise to ensure the output are remaining to be integers.

An *integer (or discrete) Laplacian distribution* [57] with parameter σ is a discrete distribution on $\mathbb{Z}$ with probability density function

$$f(z) = \frac{1-p_\sigma}{1+p_\sigma} \exp\left(-|z|/\sigma\right), \quad z \in \mathbb{Z},$$

where $p_\sigma = \exp(-1/\sigma)$.

It is easy to check a random variable $Z \sim \mathrm{Lap}_{\mathbb{Z}}(\sigma)$ is mean-zero and sub-exponential with variance $\mathrm{Var(Z)} \leq 2\sigma^2$. Moreover, Lemma 2.5 still holds if the domain of input data is a subset of $\mathbb{Z}$ and we change the noises to be integer Laplacian random variables with the same parameters. Therefore, for simplicity, we abuse the notation and use "Lap($\cdot$)" for both cases.

Chapter 3

Private Measure Mechanism

In this chapter, we will discuss the general case of synthetic data generation: given any input data $\mathcal{X} = \{X_1, \ldots, X_n\}$ from a metric space (Ω, ρ), building an algorithm that output private synthetic data with some accuracy guarantees in 1-Wasserstein distance. In Section 3.1 and Section 3.2, we will introduce two algorithms: PSMM and PMM, respectively. In particular, we focus on the behavior of the algorithms under the case of $([0,1]^d, \|\cdot\|_\infty)$.

3.1 Private signed measure mechanism (PSMM)

We will first discuss Private signed measure mechanism (PSMM), which is an easier and more intuitive approach. The procedure of PSMM is formally described in Algorithm 1.

Note that in the *output* step of Algorithm 1, the size of the synthetic data m' depends on the rational approximation of the density function of $\hat{\nu}$, and we discuss the details here. Let $\hat{v}_1, \ldots, \hat{v}_m$ be the weight of the probability measure $\hat{\nu}$ on $y_1, \ldots, y_m$, respectively. We can choose rational numbers $r_1, \ldots, r_m$ such that $\max_{i\in[m]} |r_i - \hat{\nu}_i|$ is arbitrarily small. Let m' be the least common multiple of the denominators of $r_1, \ldots, r_m$, then we output the synthetic

dataset $\hat{\mathcal{Y}}$ containing $m'r_i$ copies of y_i for $i = 1, \ldots, m$.

Algorithm 1 Private Signed Measure Mechanism

Input: true data $\mathcal{X} = (x_1, \ldots, x_n) \in \Omega^n$, partition $(\Omega_1, \ldots, \Omega_m)$ of Ω, privacy parameter $\varepsilon > 0$.

(Compute the true counts) Compute the true count in each regime $n_i = \#\{x_j \in \Omega_i : j \in [n]\}$.

(Create a new dataset) Choose any element $y_i \in \Omega_i$ independently of $\mathcal{X}$, and let $\mathcal{Y}$ be the collection of n_i copies of y_i for each $i \in [n]$.

(Add noise) Perturb the empirical measure $\mu_{\mathcal{Y}}$ of $\mathcal{Y}$ and obtain a signed measure ν such that

$$\nu(\{y_i\}) := (n_i + \lambda_i)/n,$$

where $\lambda_i \sim \mathrm{Lap}_{\mathbb{Z}}(1/\varepsilon)$ are i.i.d. discrete Laplacian random variables.

(Linear programming) Find the closest probability measure $\hat{\nu}$ of ν in d_{BL}-metric using Algorithm 2, and generate synthetic data $\widehat{\mathcal{Y}}$ from $\hat{\nu}$.

Output: synthetic data $\widehat{\mathcal{Y}} = (y_1, \ldots, y_{m'}) \in \Omega^{m'}$ for some integer m'.

Algorithm 2 Linear programming for d_{BL}-closest probability measure

Input: A discrete signed measure ν supported on $\mathcal{Y} = \{y_1, \ldots, y_m\}$.

(Compute the distances) Compute the pairwise distances $\{\|y_i - y_j\|_\infty, i > j\}$.

(Solve the linear programming) Solve the linear programming problem with $2m^2$ variables and $m + 1$ constraints:

$$\begin{aligned}
\text{minimize} \quad & \sum_{i,j=1}^{m} \|y_i - y_j\|_\infty (u_{ij} + u'_{ij}) + 2\, v_i \\
\text{s.t.} \quad & \sum_{j=1}^{m} (u_{ij} - u'_{ij}) + v_i + \tau_i \geq \nu(\{y_i\}), && \forall i \leq m, \\
& \sum_{i=1}^{m} \tau_i = 1, \\
& u_{ij}, u'_{ij}, v_i, \tau_i \geq 0, && \forall i, j \leq m, i \neq j.
\end{aligned}$$

Output: a probability measure $\hat{\nu}$ with $\hat{\nu}(\{y_i\}) = \tau_i$.

3.1.1 Laplacian complexity

Before analyzing the privacy and accuracy of PSMM, we introduce a useful complexity measure of a given function class, which quantifies the influence of the Laplacian noise on the function class.

Given the Kantorovich-Rubinstein duality (2.1), to control the W_1-distance between the original measure and the private measure, we need to describe how Lipschitz functions behave under Laplacian noise. As an analog of the worst-case Rademacher complexity [9, 45], we consider the worst-case Laplacian complexity. Such a worst-case complexity measure appears since the original dataset is deterministic without any distribution assumption.

Definition 3.1 (Worst-case Laplacian complexity)**.** *Let $\mathcal{F}$ be a function class on a metric space Ω. The worst-case Laplacian complexity of $\mathcal{F}$ is defined by*

$$L_n(\mathcal{F}) := \sup_{X_1,\dots,X_n\in\Omega} \mathbb{E}\left[\sup_{f\in\mathcal{F}}\left|\frac{1}{n}\sum_{i=1}^{n}\lambda_i f(X_i)\right|\right], \tag{3.1}$$

where $\lambda_1,\dots,\lambda_n \sim \mathrm{Lap}(1)$ are i.i.d. random variables.

Since Laplacian random variables are sub-exponential but not sub-gaussian, its complexity measure is not equivalent to the Gaussian or Rademacher complexity, but it is related to the suprema of the mixed tail process [30] and the quadratic empirical process [76]. Our next proposition bounds $L_n(\mathcal{F})$ in terms of the covering numbers of $\mathcal{F}$. Its proof is a classical application of Dudley's chaining method (see, e.g., [92]).

Proposition 3.2 (Bounding Laplacian complexity with Dudley's entropy integral)**.** *Suppose that (Ω,ρ) is a metric space and $\mathcal{F}$ is a set of functions on Ω. Then*

$$L_n(\mathcal{F}) \le C \inf_{\alpha>0}\left(2\alpha + \frac{1}{\sqrt{n}}\int_\alpha^\infty \sqrt{\log\mathcal{N}(\mathcal{F},u,\|\cdot\|_\infty)}\mathrm{d}u + \frac{1}{n}\int_\alpha^\infty \log\mathcal{N}(\mathcal{F},u,\|\cdot\|_\infty)\mathrm{d}u\right)$$

where $\mathcal{N}(\mathcal{F}, u, \|\cdot\|_\infty)$ is the covering number of $\mathcal{F}$ and $C > 0$ is an absolute constant.

Proof. We will apply the chaining argument (see, e.g., [92, Chapter 8]) to deduce a bound similar to Dudley's inequality.

Step 1: (Finding nets)

Define $\varepsilon_j = 2^{-j}$ for $j \in \mathbb{Z}$ and consider an ε_j-net T_j of $\mathcal{F}$ of size $\mathcal{N}(\mathcal{F}, \varepsilon_j, \|\cdot\|_\infty)$. Then for any $f \in \mathcal{F}$ and any level j, we can find the closest element in the net, denoted $\pi_j(f)$. In other words, there exists $\pi_j(f)$ s.t.

$$\pi_j(f) \in T_j, \quad \|f - \pi_j(f)\|_\infty \le \varepsilon_j.$$

Let m be a positive integer to be determined later, we have the telescope sum together with triangle inequality

$$\begin{aligned}\mathbb{E} \sup_{f\in\mathcal{F}} \frac{1}{n} \left| \sum_{i=1}^n f(X_i)\lambda_i \right| \le \mathbb{E} \sup_{f\in\mathcal{F}} \frac{1}{n} \left| \sum_{i=1}^n \big(f - \pi_m(f)\big)(X_i) \cdot \lambda_i \right| \\ + \sum_{j=j_0+1}^m \mathbb{E} \sup_{f\in\mathcal{F}} \frac{1}{n} \left| \sum_{i=1}^n \big(\pi_j(f) - \pi_{j-1}(f)\big)(X_i) \cdot \lambda_i \right|.\end{aligned}$$

Note that when $j = j_0$ is small enough, Ω can be covered by $\pi_{j_0}(f) \equiv 0$.

Step 2: (Bounding the telescoping sum)

For a fixed $j_0 < j \le m$, we consider the quantity

$$\mathbb{E} \sup_{f\in\mathcal{F}} \frac{1}{n} \left| \sum_{i=1}^n \big(\pi_j(f) - \pi_{j-1}(f)\big)(X_i) \cdot \lambda_i \right|.$$

For simplicity we will denote $a_i = a_i(f)$ as the coefficient $\frac{1}{n}\big(\pi_j(f) - \pi_{j-1}(f)\big)(X_i)$. Then we

have

$$|a_i| \le \frac{1}{n}\left\|f - \pi_{j-1}(f)\right\|_\infty + \frac{1}{n}\left\|\pi_j(f) - f\right\|_\infty \le \frac{1}{n}(\varepsilon_j + \varepsilon_{j-1}) \le \frac{3\varepsilon_j}{n}.$$

Since $\{\lambda_i\}_{i\in[n]}$ are independent subexponential random variables, we can apply Bernstein's inequality to the sum $\sum_i a_i\lambda_i$. Let $K = 3\varepsilon_j$, we have

$$\begin{aligned}
\mathbb{P}\left\{\left|\sum_{i=1}^n a_i\lambda_i\right| > t\right\} &\le 2\exp\left[-c\min\left(\frac{t^2}{\|a\|_2^2}, \frac{t}{\|a\|_\infty}\right)\right] \\
&\le 2\exp\left[-c\min\left(\frac{t^2}{K^2/n}, \frac{t}{K/n}\right)\right] \\
&= 2\exp\left[-cn\min\left(\frac{t^2}{K^2}, \frac{t}{K}\right)\right],
\end{aligned}$$

Then we can use the union bound to control the supreme. Define $N = |T_j| \cdot |T_{j-1}| \le |T_j|^2$,

$$\begin{aligned}
\mathbb{P}\left\{\sup_{f\in\mathcal{F}}\left|\sum_{i=1}^n a_i\lambda_i\right| > t\right\} &\le 2N\exp\left[-cn\min\left(\frac{t^2}{K^2}, \frac{t}{K}\right)\right] \wedge 1 \\
&= 2\exp\left[\log N - cn\min\left(\frac{t^2}{K^2}, \frac{t}{K}\right)\right] \wedge 1 \\
&\le 2\exp\left(\log N - cn\frac{t^2}{K^2}\right) \wedge 1 + 2\exp\left(\log N - cn\frac{t}{K}\right) \wedge 1
\end{aligned}$$

and hence

$$\begin{aligned}
\mathbb{E}\sup_{f\in\mathcal{F}}\left|\sum_{i=1}^n a_i\lambda_i\right| &= \int_0^\infty 2\exp\left(\log N - cn\frac{t^2}{K^2}\right) \wedge 1\mathrm{d}t \\
&\quad + \int_0^\infty 2\exp\left(\log N - cn\frac{t}{K}\right) \wedge 1\mathrm{d}t \\
&=: I_2 + I_1.
\end{aligned}$$

Compute them separately.

$$
\begin{aligned}
I_1 &= \int_0^\infty 2\exp\left(\log N - cn\frac{t}{K}\right) \wedge 1 \mathrm{d}t \\
&= \frac{K\log N}{cn} + \int_{K\log N/cn}^\infty 2\exp\left(\log N - cn\frac{t}{K}\right) \\
&= \frac{K\log N}{cn} + \int_0^\infty 2\exp\left(-cn\frac{t}{K}\right) \\
&\leq CK\frac{\log N}{n}
\end{aligned}
$$

$$
\begin{aligned}
I_2 &= \int_0^\infty 2\exp\left(\log N - cn\frac{t^2}{K^2}\right) \wedge 1 \mathrm{d}t \\
&= \sqrt{\frac{K^2\log N}{cn}} + \int_{\sqrt{K^2\log N/cn}}^\infty 2\exp\left(\log N - cn\frac{t^2}{K^2}\right) \\
&= \sqrt{\frac{K^2\log N}{cn}} + \int_0^\infty 2\exp\left(-cn\frac{t^2}{K^2} - 2\sqrt{cn\log N}\frac{t}{K}\right) \\
&\leq \sqrt{\frac{K^2\log N}{cn}} + \frac{K}{\sqrt{cn\log N}} \\
&\leq CK\sqrt{\frac{\log N}{n}}.
\end{aligned}
$$

Therefore we concluded that for a fixed j,

$$
\mathbb{E}\sup_{f\in\mathcal{F}}\left|\sum_{i=1}^n a_i\lambda_i\right| \leq CK\left(\frac{\log N}{n} + \sqrt{\frac{\log N}{n}}\right) \lesssim \varepsilon_j\left(\frac{\log N}{n} + \sqrt{\frac{\log N}{n}}\right)
$$

Step 3: (Bounding the last entry)

For the last entry in the telescoping sum, similarly, we denote $a_i := \frac{1}{n}\left(f - \pi_m(f)\right)(X_i)$ and

we have $|a_i| \le \varepsilon_m/n$. Then

$$\sup_{f\in\mathcal{F}}\left|\sum_{i=1}^n a_i\lambda_i\right| \le \frac{\varepsilon_m}{n}\sum_{i=1}^n |\lambda_i|,$$

and the expectation satisfies

$$\mathbb{E}\sup_{f\in\mathcal{F}}\left|\sum_{i=1}^n a_i\lambda_i\right| \le \frac{\varepsilon_m}{n}\sum_{i=1}^n \mathbb{E}\,|\lambda_i| \lesssim \varepsilon_m.$$

Step 4: (Combining the bound and choosing m) Combining the two integrals together, we deduce that for any $X_1,\dots,X_n \in \Omega$,

$$\mathbb{E}\sup_{f\in\mathcal{F}}\frac{1}{n}\left|\sum_{i=1}^n f(X_i)\lambda_i\right| \le C\left(\varepsilon_m + \sum_{j=j_0+1}^m \varepsilon_j\left(\frac{\log\mathcal{N}(\mathcal{F},\varepsilon_j,\|\cdot\|_\infty)}{n} + \sqrt{\frac{\log\mathcal{N}(\mathcal{F},\varepsilon_j,\|\cdot\|_\infty)}{n}}\right)\right).$$

Then for any fixed $\alpha > 0$, we can always choose m such that $2\alpha \le \varepsilon_m < 4\alpha$ and bound the sum above with integral

$$\mathbb{E}\sup_{f\in\mathcal{F}}\frac{1}{n}\left|\sum_{i=1}^n f(X_i)\lambda_i\right| \le C\left(2\alpha + \frac{1}{\sqrt{n}}\int_\alpha^\infty \sqrt{\log\mathcal{N}(\mathcal{F},u,\|\cdot\|_\infty)}\mathrm{d}u + \frac{1}{n}\int_\alpha^\infty \log\mathcal{N}(\mathcal{F},u,\|\cdot\|_\infty)\mathrm{d}u\right). \tag{3.2}$$

Taking infimum over α completes the proof of the first inequality.

□

In particular, we are interested in the case where $\mathcal{F}$ is the class of all the bounded Lipschitz functions. One can find the result in [88] or more explicit bound in [47] of the covering

number of the Lipschitz function class. As a result, we have the following corollary.

Corollary 3.3 (Laplacian complexity for Lipschitz functions on the hypercube)**.** *Let* $\Omega = [0,1]^d$ *with the* $\|\cdot\|_\infty$ *metric, and* $\mathcal{F}$ *be the set of all Lipschitz functions* f *on* Ω *with* $\|f\|_{\text{Lip}} \leq 1$*. We have*

$$L_n(\mathcal{F}) \leq \begin{cases} Cn^{-1/2} & \text{if } d = 1, \\ C \log n \cdot n^{-1/2} & \text{if } d = 2, \\ Cd^{-1}n^{-1/d} & \text{if } d \geq 3. \end{cases}$$

Proof. For $\Omega = [0,1]^d$ with l_∞-norm, we have $\text{diam}(\Omega) = 1$ and the covering number

$$\mathcal{N}([0,1]^d, u, \|\cdot\|_\infty) \leq u^{-d}.$$

Then, as the domain $\Omega = [0,1]^d$ is connected and centered, we can apply the bound for the covering number of $\mathcal{F}$ from [95, Theorem 17]:

$$\mathcal{N}(\mathcal{F}, u, \|\cdot\|_\infty) \leq \left(2\left\lceil 2/u \right\rceil + 1\right) 2^{\mathcal{N}\left([0,1]^d, u/2, \|\cdot\|_\infty\right)},$$

$$\implies \log \mathcal{N}(\mathcal{F}, u, \|\cdot\|_\infty) \lesssim \mathcal{N}(\Omega, u/2, \|\cdot\|_\infty) \lesssim (u/2)^{-d}.$$

Applying the inequality above to (3.2), we get

$$L_n \leq C\left(2\alpha + \frac{1}{\sqrt{n}} \int_\alpha^\infty (u/2)^{-d/2} \mathrm{d}u + \frac{1}{n} \int_\alpha^\infty (u/2)^{-d} \mathrm{d}u\right). \tag{3.3}$$

Compute the integral for the case $d = 2$ and $d \geq 3$,

$$L_n(f) \leq \begin{cases} C\left(2\alpha + \dfrac{2}{\sqrt{n}} \log \dfrac{2}{\alpha} + \dfrac{2}{n}\left(\dfrac{\alpha}{2}\right)^{-1}\right) & \text{if } d = 2. \\ C\left(2\alpha + \dfrac{2}{\sqrt{n}} \cdot \dfrac{1}{\frac{d}{2} - 1}\left(\dfrac{\alpha}{2}\right)^{1-\frac{d}{2}} + \dfrac{2}{n} \cdot \dfrac{1}{d-1}\left(\dfrac{\alpha}{2}\right)^{1-d}\right) & \text{if } d \geq 3. \end{cases}$$

Choosing $\alpha = 2n^{-1/d}$ finishes the cases for $d \geq 2$.

When $d = 1$, the Dudley integral in (3.3) is divergent. However, note that $\operatorname{diam}(\mathcal{F}) \leq 2$ and hence $\log \mathcal{N}(\mathcal{F}, u, \|\cdot\|_\infty) = 0$ for $u > 1$. From (3.2), we have

$$\begin{aligned} L_n(\mathcal{F}) &\leq C\left(2\alpha + \frac{1}{\sqrt{n}}\int_\alpha^1 (u/2)^{-1/2}\mathrm{d}u + \frac{1}{n}\int_\alpha^1 (u/2)^{-1}\mathrm{d}u\right) \\ &\leq C\left(2\alpha + \frac{2(\sqrt{2}-\sqrt{\alpha})}{\sqrt{n}} + \frac{2}{n}\log\frac{1}{\alpha}\right). \end{aligned}$$

The optimal choice of α is $\alpha \sim n^{-1/2}$, which gives us the result for $d = 1$. □

Discrete Laplacian complexity Laplacian complexity can be useful for differential privacy algorithms based on the Laplacian mechanism [34]. However, since PSMM perturbs counts in each subregion, it is more convenient for us to add integer noise to the true counts. Instead, we will use the *worst-case discrete Laplacian complexity* defined below:

$$\widetilde{L}_n(\mathcal{F}) := \sup_{X_1,\ldots,X_n\in\Omega} \mathbb{E}\left[\sup_{f\in\mathcal{F}}\left|\frac{1}{n}\sum_{i=1}^n \lambda_i f(X_i)\right|\right], \tag{3.4}$$

where $\lambda_1, \ldots, \lambda_n \sim \mathrm{Lap}_{\mathbb{Z}}(1)$ are i.i.d. discrete Laplacian random variables.

In particular, $\mathrm{Lap}_{\mathbb{Z}}(1)$ has a bounded sub-exponential norm, therefore the proof of Proposition 3.2 works for discrete Laplacian random variables as well. Consequently, Corollary 3.3 also holds for $\widetilde{L}_n(\mathcal{F})$, with a different absolute constant C.

3.1.2 Privacy and Accuracy of Algorithm 1

The privacy guarantee of Algorithm 1 can be proved by checking the definition. The essence of the proof is the same as the classical Laplacian mechanism [34].

Proposition 3.4 (Privacy of Algorithm 1)**.** *Algorithm 1 is ε-differentially private.*

Proof. It suffices to prove that the steps from $\mathcal{X}$ to the sign measure ν in Algorithm 1 is ε-differentially private since the remaining steps are only based on ν. Notice that both $\mu_{\mathcal{Y}}, \nu$ are supported on $Y_1, \ldots, Y_m$, we can identify the two discrete measures as m dimensional vectors in the standard simplex, denoted $\overline{\mu_{\mathcal{Y}}}, \overline{\nu}$, respectively. Consider two data sets $\mathcal{X}_1$ and $\mathcal{X}_2$ differ in one point. Suppose we deduced $\mu_{\mathcal{Y}_1}, \mu_{\mathcal{Y}_2}$ and ν_1, ν_2 through the first four steps of Algorithm 1 from $\mathcal{X}_1, \mathcal{X}_2$, respectively. We know two vectors $\overline{\mu_{\mathcal{Y}_1}}, \overline{\mu_{\mathcal{Y}_2}}$ are different at one coordinate, where the difference is bounded by $1/n$.

Then

$$\begin{aligned}\frac{\mathbb{P}\{\nu_1 = \eta\}}{\mathbb{P}\{\nu_2 = \eta\}} &= \prod_{i=1}^{m} \frac{\mathbb{P}\left\{\lambda_i = n(\eta - \overline{\mu_{\mathcal{Y}_1}})_i\right\}}{\mathbb{P}\left\{\lambda_i = n(\eta - \overline{\mu_{\mathcal{Y}_2}})_i\right\}} = \prod_{i=1}^{m} \frac{\exp(-\varepsilon n|(\eta - \overline{\mu_{\mathcal{Y}_1}})_i|)}{\exp(-\varepsilon n|(\eta - \overline{\mu_{\mathcal{Y}_2}})_i|)} \\ &\leq \exp\left(\varepsilon n \|\mu_{\mathcal{Y}_2} - \mu_{\mathcal{Y}_1}\|_1\right) \leq e^{\varepsilon}.\end{aligned}$$

By writing $\mathbb{P}\{\nu_i \in S\} = \sum_{\eta \in S} \mathbb{P}\{\nu_i = \eta\}$ for $i = 1, 2$, the inequality above implies Algorithm 1 is ε-differentially private. □

We now turn to accuracy. The linear programming problem stated in Algorithm 2 has $(2m^2 + 2m)$ many variables and $(m+1)$ many constraints, which can be solved in polynomial time in m. We first show that Algorithm 2 indeed outputs the closest probability measure to ν in the d_{BL}-distance in the next proposition.

Proposition 3.5. *For a discrete signed measure ν on Ω, Algorithm 2 gives its closest probability measure in d_{BL}-distance with the same support set with a polynomial running time in m.*

Proof. For two signed measures τ, ν supported on $\mathcal{Y}$, the d_{BL}-distance between τ and ν is

$$d_{\mathrm{BL}}(\tau, \nu) = \sup_{\|f\|_{\mathrm{Lip}} \leq 1} \left| \sum_{i=1}^{m} f(y_i) \left(\tau(\{y_i\}) - \nu(\{y_i\}) \right) \right|.$$

For simplicity, we denote $f_i = f(y_i)$, $\nu_i = \nu(\{y_i\})$ and $\tau_i = \tau(\{y_i\})$. Then we note that for any f with $\|f\|_{\mathrm{Lip}} \leq 1$, only $(f_i)_{i \in [m]}$ matters in the definition above. Therefore, suppose ν and τ are fixed, computing the d_{BL}-distance is equivalent to the following linear programming problem:

$$\begin{aligned}
\max \quad & \sum_{i=1}^{m} (\nu_i - \tau_i) f_i \\
\text{s.t.} \quad & f_i - f_j \leq \|y_i - y_j\|_\infty, && \forall i, j \leq m, i \neq j, \\
& -f_i + f_j \leq \|y_i - y_j\|_\infty, && \forall i, j \leq m, i \neq j, \\
& -1 \leq f_i \leq 1, && \forall i \leq m.
\end{aligned}$$

After a change of variable $f_i' = f_i + 1$, we can rewrite it as

$$\begin{aligned}
\max \quad & \sum_{i=1}^{m} (\nu_i - \tau_i) f_i' - \left(\nu(\Omega) - 1 \right) \\
\text{s.t.} \quad & f_i' - f_j' \leq \|y_i - y_j\|_\infty, && \forall i, j \leq m, i \neq j, \\
& -f_i' + f_j' \leq \|y_i - y_j\|_\infty, && \forall i, j \leq m, i \neq j, \\
& 0 \leq f_i' \leq 2, && \forall i \leq m.
\end{aligned}$$

Next, we can consider the dual problem of the linear programming problem above. The duality theory in linear programming [91, Chapter 12] showed that the original problem and the dual problem have the same optimal solution. Let $u_{ij}, u_{ij}' \geq 0$ be the dual variable for

the linear constraints about $f'_i - f'_j$ and $-f'_i + f'_j$, and let $v_i \geq 0$ be the dual variable for the equation $f'_i \leq 2$. As the linear programming above is in the standard form, by the duality theory, it is equivalent to

$$\begin{aligned}
\min \quad & \sum_{i \neq j} \|y_i - y_j\|_\infty (u_{ij} + u'_{ij}) + 2\, v_i - \big(\nu(\Omega) - 1\big) \\
\text{s.t.} \quad & \sum_{j \neq i} (u_{ij} - u'_{ij}) + v_i \geq \nu_i - \tau_i, && \forall i \leq m, \\
& u_{ij}, u'_{ij}, v_i \geq 0 && \forall i, j \leq m, i \neq j.
\end{aligned}$$

To find the minimizer τ for a given ν, we regard τ_i as variables and add the constraints of τ being a probability measure. Also, we can eliminate the constant $\nu(\Omega) - 1$ in the target function. So we get the linear programming problem:

$$\begin{aligned}
\min \quad & \sum_{i \neq j} \|y_i - y_j\|_\infty (u_{ij} + u'_{ij}) + 2\, v_i \\
\text{s.t.} \quad & \sum_{j \neq i} (u_{ij} - u'_{ij}) + v_i + \tau_i \geq \nu_i, && \forall i \leq m, \\
& \sum_{i=1}^{m} \tau_i = 1, \\
& u_{ij}, u'_{ij}, v_i, \tau_i \geq 0 && \forall i, j \leq m, i \neq j.
\end{aligned}$$

There are $2m^2$ variables in total and $m + 1$ linear constraints, and the minimizer $(\tau_i)_{i=1}^m$ is what we want. □

Now we are ready to analyze the accuracy of Algorithm 1. In PSMM, independent Laplacian noise is added to the count of each sub-region. Therefore, the Laplacian complexity arises when considering the expected Wasserstein distance between the original empirical measure and the synthetic measure.

Theorem 3.6 (Accuracy of Algorithm 1)**.** *Suppose* $(\Omega_1, \ldots, \Omega_m)$ *is a partition of* (Ω, ρ) *and* $\mathcal{F}$ *is the set of all functions with Lipschitz norm bounded by* 1*. Then the measure* $\hat{\nu}$ *generated from Algorithm 1 satisfies*

$$\mathbb{E}\, W_1(\mu_{\mathcal{X}}, \hat{\nu}) \leq \max_i \operatorname{diam}(\Omega_i) + \frac{2m}{\varepsilon n} \widetilde{L}_m(\mathcal{F}).$$

Proof. We transformed the original data measure $\mu_{\mathcal{X}}$ with three steps: $\mu_{\mathcal{X}} \longrightarrow \mu_{\mathcal{Y}} \longrightarrow \nu \longrightarrow \hat{\nu}$.

Step 1: For the first step in the algorithm, we have $W_1(\mu_{\mathcal{X}}, \mu_{\mathcal{Y}}) \leq \max_i \operatorname{diam}(\Omega_i)$. This follows from the definition of 1-Wasserstein distance.

Step 2: In this step, ν is no longer a probability measure, and we consider $d_{\mathrm{BL}}(\mu_{\mathcal{Y}}, \nu)$ instead:

$$\begin{aligned} \mathbb{E} d_{\mathrm{BL}}(\mu_{\mathcal{Y}}, \nu) &= \mathbb{E} \sup_{\|f\|_{\mathrm{Lip}} \leq 1} \left| \int f \mathrm{d}\mu_{\mathcal{Y}} - \int f \mathrm{d}\nu \right| \\ &= \mathbb{E} \sup_{\|f\|_{\mathrm{Lip}} \leq 1} \left| \sum_{i=1}^{m} f(y_i) \left(\frac{n_i}{n} + \frac{\lambda_i}{n} - \frac{n_i}{n} \right) \right| = \frac{m}{\varepsilon n} \widetilde{L}_m(\mathcal{F}), \end{aligned} \tag{3.5}$$

where $\mathcal{F}$ is the function class of f with $\|f\|_{\mathrm{Lip}}$

Step 3: For the last step, we have $d_{\mathrm{BL}}(\nu, \hat{\nu}) \leq d_{\mathrm{BL}}(\mu_{\mathcal{Y}}, \nu)$ because $\hat{\nu}$ is the closest probability measure to ν from Proposition 3.5. As a result, we have

$$\begin{aligned} W_1(\mu_{\mathcal{X}}, \hat{\nu}) = d_{\mathrm{BL}}(\mu_{\mathcal{X}}, \hat{\nu}) &\leq d_{\mathrm{BL}}(\mu_{\mathcal{X}}, \mu_{\mathcal{Y}}) + d_{\mathrm{BL}}(\mu_{\mathcal{Y}}, \nu) + d_{\mathrm{BL}}(\nu, \hat{\nu}) \\ &\leq W_1(\mu_{\mathcal{X}}, \mu_{\mathcal{Y}}) + 2 d_{\mathrm{BL}}(\mu_{\mathcal{Y}}, \nu) \\ &\leq \max_i \operatorname{diam}(\Omega_i) + 2 d_{\mathrm{BL}}(\mu_{\mathcal{Y}}, \nu). \end{aligned}$$

After taking the expectation, we can apply (3.5) to get the desired inequality. □

Note that $\operatorname{diam}(\Omega_i) \asymp m^{-1/d}$ can be satisfied when we take a partition of $\Omega = [0,1]^d$ where each Ω_i is a subcube of the same size. Using the formula above and the result of Laplacian complexity for the hypercube in Corollary 3.3, one can easily deduce the following result.

Corollary 3.7 (Accuracy of Algorithm 1 on the hypercube)**.** *Take* $m = \lceil \varepsilon n \rceil$ *and let* $(\Omega_1, \dots, \Omega_m)$ *be a partition of* $\Omega = [0,1]^d$ *with the norm* $\|\cdot\|_\infty$*. Assume that* $\operatorname{diam}(\Omega_i) \asymp m^{-1/d}$*. Then the measure* $\hat{\nu}$ *generated from Algorithm 1 satisfies*

$$\mathbb{E}\, W_1(\mu_{\mathcal{X}}, \hat{\nu}) \leq \begin{cases} C(\varepsilon n)^{-\frac{1}{2}} & \text{if } d = 1, \\ C \log(\varepsilon n)(\varepsilon n)^{-\frac{1}{2}} & \text{if } d = 2, \\ C(\varepsilon n)^{-\frac{1}{d}} & \text{if } d \geq 3. \end{cases}$$

Proof. Using Theorem 3.6, we have

$$\mathbb{E}\, W_1(\mu_{\mathcal{X}}, \hat{\nu}) \leq \max_i \operatorname{diam}(\Omega_i) + \frac{2m}{\varepsilon n} \widetilde{L}_m(\mathcal{F}).$$

By assumption we have $\max_i \operatorname{diam}(\Omega_i) \asymp m^{-1/d} \asymp (\varepsilon n)^{-1/d}$. And by 3.3 we have the bound for the Laplacian complexity

$$\widetilde{L}_m(\mathcal{F}) \leq \begin{cases} C(\varepsilon n)^{-1/2} & \text{if } d = 1, \\ C \log n \cdot (\varepsilon n)^{-1/2} & \text{if } d = 2, \\ C(\varepsilon n)^{-1/d} & \text{if } d \geq 3. \end{cases}$$

When $d \geq 3$, the two terms are comparable. And when $d = 1, 2$, the Laplacian complexity dominates the error. Combining the two inequalities gives the result. □

3.2 Private measure mechanism (PMM)

3.2.1 Binary partition and noisy counts

In the PMM algorithm, we continue the methodology of partitioning the domain Ω and perturbing the count in each subregion with independent noises to ensure privacy. Recall that PSMM from Section 3.1 only requires the diameters of the subregions to be small enough to guarantee accuracy. In this section, we will show that applying the binary hierarchical partition in Definition 2.7 will reduce the 1-Wasserstein error as well as the time complexity of the algorithm.

Let $(\Omega_\theta)_{\theta\in\{0,1\}^{\leq r}}$ be a binary partition of Ω as proposed in Definition 2.7. Given true data $(X_1,\dots,X_n)\in\Omega^n$, the *true count* n_θ is the number of data points in the region Ω_θ, i.e.

$$n_\theta \coloneqq \left|\left\{i\in[n]:\ X_i\in\Omega_\theta\right\}\right|.$$

We will convert true counts into *noisy counts* n'_θ by adding Laplacian noise; all regions on the same level will receive noise of the same expected magnitude. Formally, we set

$$n'_\theta \coloneqq (n_\theta+\lambda_\theta)_+, \quad \text{where} \quad \lambda_\theta\sim \mathrm{Lap}_{\mathbb{Z}}(\sigma_j),$$

and $j\in\{0,\dots,r\}$ is the level of θ. At this point, the magnitudes of the noise σ_j can be arbitrary.

3.2.2 Consistency

The true counts n_θ are non-negative and *consistent*, i.e., the counts of subregions always add up to the count of the region:

$$n_{\theta 0} + n_{\theta 1} = n_\theta \quad \text{for all } \theta \in \{0,1\}^{\leq r-1}.$$

The noisy counts n'_θ are non-negative, but not necessarily consistent. Algorithm 3 enforces consistency by adjusting the counts iteratively, from top to bottom. In the case of the deficit, when the sum of the two subregional counts is smaller than the count of the region, we increase both subregional counts. In the opposite case or surplus, we decrease both subregional counts. Apart from this requirement, we are free to distribute the deficit or surplus between the subregional counts.

It is convenient to state this requirement by considering a *product partial order* on $\mathbb{Z}_+^2$, where we declare that $(a_0, a_1) \preceq (b_0, b_1)$ if and only if $a_0 \leq b_0$ and $a_1 \leq b_1$. We call the two vectors $a, b \in \mathbb{Z}^2$ *comparable* if either $a \preceq b$ or $b \preceq a$. Furthermore, $L(a)$ denotes the line $x + y = a$ on the plane.

Algorithm 3 Consistency

Input: non-negative numbers $(n'_\theta)_{\theta \in \{0,1\}^{\leq r}}$, where n' is a nonnegative integer.
set $m \coloneqq n'$.
for $j = 0, \ldots, r-1$ **do**
  for $\theta \in \{0,1\}^j$ **do**
    transform the vector $(n'_{\theta 0}, n'_{\theta 1}) \in \mathbb{Z}_+^2$ into any comparable vector $(m_{\theta 0}, m_{\theta 1}) \in \mathbb{Z}_+^2 \cap L(m_\theta)$.
  end for
end for
Output: non-negative integers $(m_\theta)_{\theta \in \{0,1\}^{\leq r}}$.

At each step, Algorithm 3 uses a transformation $f_\theta : \mathbb{Z}_+^2 \to \mathbb{Z}_+^2 \cap L(m_\theta)$. It can be chosen arbitrarily; the only requirement is that $f_\theta(x)$ be comparable with x. The comparability requirement is natural and non-restrictive. For example, the *uniform* transformation selects

the closest point in the discrete interval $\mathbb{Z}_+^2 \cap L(m_\theta)$ in (say) the Euclidean metric. Alternatively, the *proportional* transformation selects the point in the discrete interval $\mathbb{Z}_+^2 \cap L(m_\theta)$ that is closest to the line that connects the input vector and the origin.

3.2.3 PMM algorithm and its privacy and accuracy

Algorithm 3 ensures that its output counts m_θ are non-negative, integer, and consistent. They are also private since they are generated from a function of the noisy counts n'_θ, which are private as we proved. Therefore, the counts m_θ can be used to generate *private synthetic data* by putting m_θ points in cell Ω_θ. We now present Algorithm 4, *Private Measure Mechanism*.

Algorithm 4 Private Measure Mechanism

Input: true data $\mathcal{X} = (x_1, \ldots, x_n) \in \Omega^n$, noise magnitudes $\sigma_0, \ldots, \sigma_r > 0$.

(Compute true counts) Let n_θ be the number of data points in Ω_θ.

(Add noise) Let $n'_\theta := (n_\theta + \lambda_\theta)_+$, where $\lambda_\theta \sim \mathrm{Lap}_{\mathbb{Z}}(\sigma_j)$ are i.i.d. random variables,

(Enforce consistency) Convert the noisy counts (n'_θ) to consistent counts (m_θ) using Algorithm 3.

(Sample) Choose any m_θ points in each cell Ω_θ, $\theta \in \{0,1\}^r$ independently of $\mathcal{X}$.

Output: the set of all these points as synthetic data $\mathcal{Y} = (y_1, \ldots, y_m) \in \Omega^m$.

We first prove that Algorithm 4 is differentially private. The proof idea is similar to the classic Laplacian mechanism. But now our noise is of differential scale for each level, so more delicate calculations are needed.

Theorem 3.8 (Privacy of Algorithm 4)**.** *The vector of noisy counts $(n_\theta + \lambda_\theta)$ in Algorithm 4 is ε-differentially private, where*

$$\varepsilon = \sum_{j=0}^{r} \frac{1}{\sigma_j}.$$

Consequently, the synthetic data $\mathcal{Y}$ generated by Algorithm 4 is ε-differentially private.

Proof. Consider the map $F(\mathcal{X}) = (n_\theta)$ that transforms the input data into the vector of counts. Suppose a pair of input data $\mathcal{X}$ and $\tilde{\mathcal{X}}$ differ in one point x_i. Consider the corresponding vectors of counts (n_θ) and $(\tilde{n}_\theta)$. For each level $j = 0, \dots, r$, the vectors of counts differ for a single $\theta \in \{0,1\}^j$, namely for the θ that corresponds to the region Ω_θ containing x_i. Moreover, whenever such a difference occurs, we have $|n_\theta - \tilde{n}_\theta| = 1$. Thus, extending the vector $(\sigma_j)_{j=0}^r$ to $(\sigma_\theta)_{\theta \in \{0,1\}^{\le r}}$ trivially (by converting σ_j to σ_θ for all $\theta \in \{0,1\}^j$), we have

$$\left\| F(\mathcal{X}) - F(\tilde{\mathcal{X}}) \right\|_{\ell^1(\sigma)} = \sum_{j=0}^{r} \frac{1}{\sigma_j} \sum_{\theta \in \{0,1\}^j} |n_\theta - \tilde{n}_\theta| = \sum_{j=0}^{r} \frac{1}{\sigma_j} = \varepsilon.$$

Applying Lemma 2.5, we conclude that the map $\mathcal{X} \mapsto (n_\theta + \lambda_\theta)$ is ε-differentially private. □

We now turn to its accuracy. The accuracy is determined by the magnitudes of the noise σ_j and by the multiscale geometry of the domain Ω. The latter is captured by the diameters of the regions Ω_θ, specifically by their sum at each level, which we denote

$$\Delta_j \coloneqq \sum_{\theta \in \{0,1\}^j} \operatorname{diam}(\Omega_\theta) \tag{3.6}$$

and adopt the notation $\Delta_{-1} \coloneqq \Delta_0 = \operatorname{diam}(\Omega)$. In addition to Δ_j, the accuracy is affected by the *resolution* of the partition, which is the maximum diameter of the cells, denoted by

$$\delta \coloneqq \max_{\theta \in \{0,1\}^r} \operatorname{diam}(\Omega_\theta).$$

Theorem 3.9 (Accuracy of Algorithm 4)**.** *Algorithm 4 that transforms true data $\mathcal{X}$ into synthetic data $\mathcal{Y}$ has the following expected accuracy in the Wasserstein metric:*

$$\mathbb{E}\, W_1(\mu_{\mathcal{X}}, \mu_{\mathcal{Y}}) \le \frac{2\sqrt{2}}{n} \sum_{j=0}^{r} \sigma_j \Delta_{j-1} + \delta.$$

Here $\mu_{\mathcal{X}}$ and $\mu_{\mathcal{Y}}$ are the empirical probability distributions on the true and synthetic data,

respectively.

The detailed proof to Theorem 3.9 is deferred to Section 3.2.4.

The privacy and accuracy guarantees of Algorithm 4 (Theorems 3.8 and 3.9) hold for any choice of noise levels σ_j. By optimizing σ_j, we can achieve the best accuracy for a given level of privacy.

Theorem 3.10 (Optimized accuracy)**.** *With the optimal choice of magnitude levels* (3.7)*, Algorithm 4 that transforms true data $\mathcal{X}$ into synthetic data $\mathcal{Y}$ is ε-differential private, and has the following expected accuracy in the* 1*-Wasserstein distance:*

$$\mathbb{E}\, W_1\left(\mu_{\mathcal{X}}, \mu_{\mathcal{Y}}\right) \leq \frac{\sqrt{2}}{\varepsilon n}\Big(\sum_{j=0}^{r} \sqrt{\Delta_{j-1}}\Big)^2 + \delta.$$

Here $\mu_{\mathcal{X}}$ and $\mu_{\mathcal{Y}}$ are the empirical measures of the true and synthetic data, respectively.

Proof. We will use the Lagrange multipliers procedure to find the optimal choices of σ_j. Given the maximal layer r, recall Theorem 3.8, we should use our privacy budget as

$$\varepsilon = \sum_{j=0}^{r} \frac{1}{\sigma_j}.$$

Therefore, we aim to minimize the accuracy bound with the specified privacy budget, namely

$$\text{minimize}\, \mathbb{E}\, W_1(\mu_{\mathcal{X}}, \mu_{\mathcal{Y}}) \quad \text{s.t. } \varepsilon = \sum_{j=0}^{r} \frac{1}{\sigma_j}.$$

Recall the result in Theorem 3.9. Here ε, n are given and δ is fixed as long as we determine the maximal level r. So the minimization problem is

$$\text{minimize} \sum_{j=0}^{r} \sigma_j \Delta_{j-1} \quad \text{s.t. } \varepsilon = \sum_{j=0}^{r} \frac{1}{\sigma_j}.$$

Consider the Lagrangian function

$$f(\sigma_0, \ldots, \sigma_r; t) \coloneqq \sum_{j=0}^{r} \sigma_j \Delta_{j-1} - t \left(\sum_{j=0}^{r} \frac{1}{\sigma_j} - \varepsilon \right)$$

and the corresponding equation

$$\frac{\partial f}{\partial \sigma_0} = \cdots = \frac{\partial f}{\partial \sigma_r} = \frac{\partial f}{\partial t} = 0.$$

One can easily check that the equations above have a unique solution

$$\sigma_j = \frac{S}{\varepsilon \sqrt{\Delta_{j-1}}} \quad \text{where} \quad S = \sum_{j=0}^{r} \sqrt{\Delta_{i-1}}. \tag{3.7}$$

and it is indeed a minimal point for $f(\sigma_0, \ldots, \sigma_r; t)$.

As a result, if we fix ε and want Algorithm 4 to be ε-differentially private, we should choose the noise magnitudes as (3.7). Substituting these noise magnitudes into the accuracy Theorem 3.9, we see that the accuracy gets bounded by $\frac{\sqrt{2}}{\varepsilon n} S^2 + \delta$. □

Corollary 3.11 (Optimized accuracy for hypercubes)**.** *When $\Omega = [0,1]^d$ equipped with the ℓ^∞ metric, with the optimal choice of magnitude levels* (3.7) *and the optimal choice of*

$$r = \begin{cases} \log_2(\varepsilon n) - 1 & \text{if } d = 1, \\ \log_2(\varepsilon n) & \text{if } d \geq 2, \end{cases}$$

we have

$$\mathbb{E}\, W_1(\mu_{\mathcal{X}}, \mu_{\mathcal{Y}}) \lesssim \begin{cases} \dfrac{\log^2(\varepsilon n)}{\varepsilon n}, & \text{if } d = 1, \\ (\varepsilon n)^{-1/d}, & \text{if } d \geq 2. \end{cases}$$

Proof. Let $\Omega = [0,1]$ with the ℓ^∞ metric. The natural hierarchical binary decomposition of $[0,1]$ (cut through the middle) makes subintervals of length $\operatorname{diam}(\Omega_\theta) = 2^{-j}$ for $\theta \in \{0,1\}^j$, so $\Delta_j = 1$ for all j, and the resolution is $\delta = 2^{-r}$. Theorem 3.10 makes ε-differential private synthetic data with accuracy

$$\mathbb{E}\, W_1\left(\mu_{\mathcal{X}}, \mu_{\mathcal{Y}}\right) \leq \frac{\sqrt{2}(r+1)^2}{\varepsilon n} + 2^{-r}.$$

A nearly optimal choice for r is $r = \log_2(\varepsilon n) - 1$, which yields

$$\mathbb{E}\, W_1\left(\mu_{\mathcal{X}}, \mu_{\mathcal{Y}}\right) \leq \frac{(2+\sqrt{2})\log_2^2(\varepsilon n)}{\varepsilon n}.$$

The optimal noise magnitudes, per (3.7), are $\sigma_j = \log_2^2(\varepsilon n)/\varepsilon$. In other words, the noise *does not decay* with the level.

Let $\Omega = [0,1]^d$ for $d > 1$. The natural hierarchical binary decomposition of $[0,1]^d$ (cut through the middle along a coordinate hyperplane) makes subintervals of length $\operatorname{diam}(\Omega_\theta) \asymp 2^{-j/d}$ for $\theta \in \{0,1\}^j$, so $\Delta_j = 2^j \cdot 2^{-j/d} = 2^{(1-1/d)j}$ for all j, and the resolution is $\delta = 2^{-r/d}$. Thus,

$$S = \sum_{j=0}^{r} \sqrt{\Delta_{j-1}} \sim 2^{\frac{1}{2}(1-\frac{1}{d})r}.$$

Theorem 3.10 makes a ε-differential private synthetic data with accuracy

$$\mathbb{E}\, W_1\left(\mu_{\mathcal{X}}, \mu_{\mathcal{Y}}\right) \lesssim \frac{2^{(1-\frac{1}{d})r}}{\varepsilon n} + 2^{-r/d}.$$

A nearly optimal choice for the depth of the partition is $r = \log_2(\varepsilon n)$, which yields

$$\mathbb{E}\, W_1\left(\mu_{\mathcal{X}}, \mu_{\mathcal{Y}}\right) \lesssim (\varepsilon n)^{-1/d}.$$

The optimal noise magnitudes, per (3.7), are

$$\sigma_j \sim \varepsilon^{-1} 2^{\frac{1}{2}(1-\frac{1}{d})(r-j)}.$$

Thus, the *noise decays* with the level j, becoming $O(1)$ per region for the smallest regions. □

Example 3.12 (Computational efficiency of Algorithm 4)**.** *Since a binary hierarchical partition has 2^r cells in total, the running time of Algorithm 4 is $O(2^r)$. When $\Omega = [0,1]^d$, with the same optimal choice of r in Corollary 3.11, the running time of PMM becomes $O(\varepsilon dn)$.*

3.2.4 Proof of Theorem 3.9

For the proof of Theorem 3.9, we introduce a quantitative notion for the incomparability of two vectors on the plane. For vectors $a, b \in \mathbb{Z}_+^2$, we define

$$\text{flux}(a,b) := \begin{cases} 0 & \text{if } a \text{ and } b \text{ are comparable,} \\ \min\left(|a_1 - b_1|, |a_2 - b_2|\right) & \text{otherwise.} \end{cases}$$

Lemma 3.13 (Flux as incomparability)**.** $\text{flux}(a,b)$ *is the ℓ_∞-distance from a to the set of points that are comparable to b.*

Proof. If a, b are comparable, both values are zero. If a, b is not comparable, we can assume $a_1 > b_1$, $a_2 < b_2$ without loss of generality. The set of points that are comparable to b is

$$\{(x_1, x_2) \in \mathbb{Z}_+^2 \mid x_1 \le b_1, x_2 \le b_2\} \cup \{(x_1, x_2) \in \mathbb{Z}_+^2 \mid x_1 \ge b_1, x_2 \ge b_2\}.$$

Note that the distance from a to the first set is $|a_1 - b_1|$ and the distance from a to the second set is $|a_2 - b_2|$. Then $\text{flux}(a,b)$ is the smaller one of the two distances, which is also the distance from a to the union set. □

For example, if $a = (1, 9)$ and $b = (6, 7)$, then $\text{flux}(a, b) = 2$. Note that a has a distance 2 to the vector $(1, 7)$ which is comparable with b.

Lemma 3.14 (Flux as transfer)**.** *Suppose we have two bins with a_1 and a_2 balls in them. Then one can achieve b_1 and b_2 balls in these bins by:*

(a) first making the total number of balls correct by adding a total of $(b_1 + b_2) - (a_1 - a_2)$ balls to the two bins (or removing, if that number is negative);

(b) then transferring $\text{flux}\big((a_1, a_2), (b_1, b_2)\big)$ balls from one bin to the other.

Proof. Case 1: $a = (a_1, a_2)$ and $b = (b_1, b_2)$ are comparable. If $a \preceq b$, remove $b_1 - a_1$ balls from bin 1 and $b_2 - a_2$ balls from bin 2 to achieve the result. If $b \preceq a$, adding $a_1 - b_1$ balls to bin 1 and $a_2 - b_2$ balls to bin 2 to achieve the result.

Case 2: $a = (a_1, a_2)$ and $b = (b_1, b_2)$ are incomparable. Without loss of generality, we can assume that $a_1 - b_1 \geq 0$, $a_2 - b_2 \leq 0$.

Assume first that $a_1 - b_1 \geq b_2 - a_2$. Then $\text{flux}(a, b) = b_2 - a_2 \coloneqq M$. Then $\Delta =: (a_1 + a_2) - (b_1 + b_2) > 0$. Removing Δ balls from bin 1 and transferring M balls from bin 1 to bin 2 achieves the result. Note that there are enough balls in bin 1 to transfer, since $M + \Delta = a_1 - b_1 \in [0, a_1]$.

Now assume that $a_1 - b_1 \leq b_2 - a_2$. Then $\text{flux}(a, b) = a_1 - b_1 \coloneqq M$. Then $\Delta =: (b_1 + b_2) - (a_1 + a_2) > 0$. Adding Δ balls to bin 2 and transferring M balls from bin 1 to bin 2 achieves the result. □

For example, suppose that one bin has 1 ball and the other has 9. Then we can achieve 6 and 7 balls in these bins by first adding 3 balls to the first bin and transferring 2 balls from the second to the first bin. As we noted above, 2 is the flux between the vectors $(1, 9)$ and $b = (6, 7)$.

Lemma 3.14 can be generalized to the hierarchical binary partition of Ω as follows.

Lemma 3.15. *Consider any data set $\mathcal{X} \in \Omega^n$, and let $(n_\theta)_{\theta\in\{0,1\}^r}$ be its counts. Consider any consistent vector of non-negative integers $(m_\theta)_{\theta\in\{0,1\}^r}$. Then one can transform $\mathcal{X}$ into a set $\mathcal{Z} \in \Omega^m$ that has counts $(m_\theta)_{\theta\in\{0,1\}^r}$ by:*

(a) first making the total number of points correct by adding a total of $m-n$ points to Ω (or remove, if that number is negative);

(b) then transferring flux $\big((n_{\theta 0}, n_{\theta 1}), (m_{\theta 0}, m_{\theta 1})\big)$ *points from $\Omega_{\theta 0}$ to $\Omega_{\theta 1}$ or vice versa, for all $j = 0, \ldots, r-1$ and $\theta \in \{0,1\}^j$.*

Proof. First, we make the total number of points in Ω correct by adding $m - n$ points to Ω (or removing, if that number is negative).

Apply Lemma 3.14 for the two parts of Ω: bin Ω_0 that contains n_0 points and bin Ω_1 that contains n_1 points. Since Ω already contains the correct total number of points m, we can make the two bins contain the correct number of points, i.e. m_0 and m_1 respectively, by transferring flux $\big((n_0, n_1), (m_0, m_1)\big)$ points from one bin to the other.

Apply Lemma 3.14 for the two parts of Ω_0: bin Ω_{00} that contains n_{00} points and bin Ω_{01} that contains n_{01} points. Since Ω_0 already contains the correct number of points m_0, we can make the two bins contain the correct number of points, i.e. m_{00} and m_{01} respectively, by transferring flux $\big((n_{00}, n_{01}), (m_{00}, m_{01})\big)$ points from one bin to the other.

Similarly, since Ω_1 already has the correct number of points m_1, we can make Ω_{10} and Ω_{11} contain the correct number of points m_{10} and m_{11} by transferring flux $\big((n_{00}, n_{01}), (m_{00}, m_{01})\big)$ points from one bin to the other.

Continuing this way, we can complete the proof. Note that the steps of the iteration procedure we described are interlocked. Each next step determines which subregion the transferred

points are selected from, and which subregion they are moved to in the previous step. For example, the original step calls to add (or remove) $m-n$ points to or from Ω, but does not specify how these points are distributed between the two parts Ω_0 and Ω_1. The application of Lemma 3.14 at the next step determines this. □

Combining the concept of the flux and our algorithm, the following two lemmas are useful in the proof of Theorem 3.9.

Lemma 3.16. *In Algorithm 4, we have*

$$\operatorname{flux}\left((n_{\theta 0}, n_{\theta 1}),\, (m_{\theta 0}, m_{\theta 1})\right) \leq \max\left(|\lambda_{\theta 0}|, |\lambda_{\theta 1}|\right)$$

for all $j = 0, \ldots, r-1$ *and* $\theta \in \{0,1\}^j$.

Proof. We will derive this result from Lemma 3.13. First, let us compute the distance from $a = (n_{\theta 0}, n_{\theta 1})$ to $b' = (n'_{\theta 0}, n'_{\theta 1}) = \left((n_{\theta 0} + \lambda_{\theta 0})_+,\, (n_{\theta 1} + \lambda_{\theta 1})_+\right)$. Since the map $x \mapsto x_+$ is 1-Lipschitz, we have

$$\left\|a - b'\right\|_\infty \leq \max\left(|\lambda_{\theta 0}|, |\lambda_{\theta 1}|\right).$$

Furthermore, recall that by Algorithm 3, b' is comparable to $b = (m_{\theta 0}, m_{\theta 1})$. An application of Lemma 3.13 completes the proof. □

Lemma 3.17. *For any finite multisets* $U \subset V$ *such that all elements in* U *are from* Ω*, one has*

$$W_1(\mu_U, \mu_V) \leq \frac{\left|V \setminus U\right|}{|V|} \cdot \operatorname{diam}(\Omega).$$

Proof. Finding the 1-Wasserstein distance in the discrete case is equivalent to solving the optimal transformation problem. In fact, we can obtain μ_U from μ_V by moving $\left|V \setminus U\right|$ atoms of μ_V, each having mass $1/|V|$, and distributing their mass uniformly over U. The distance

for each movement is bounded by $\operatorname{diam}(\Omega)$. Therefore the 1-Wasserstein distance between μ_U and ν_V is bounded by $\frac{|V \setminus U|}{|V|} \operatorname{diam}(\Omega)$. □

Proof of Theorem 3.9. Owing to Lemma 3.15 and Lemma 3.16, the creation of synthetic data from the true data $\mathcal{X} \mapsto \mathcal{Y}$, described by Algorithm 4, can be achieved by the following three steps.

1. Transform the n-point input set $\mathcal{X}$ to an m-point set $\mathcal{X}_1$ by adding or removing $|m-n|$ points.
2. Transform $\mathcal{X}_1$ to $\mathcal{X}_2$ by moving at most $\max\left(|\lambda_{\theta 0}|, |\lambda_{\theta 1}|\right)$ many data points for each $j = 0, 1, \ldots, r-1$ and $\theta \in \{0,1\}^j$ between the two parts of the region Ω_θ.
3. Transforms $\mathcal{X}_2$ to the output data $\mathcal{Y}$ by relocating points within their cells.

We will analyze the accuracy of these steps one at a time.

Analyzing Step 2. The total distance the points are moved at this step is bounded by

$$\sum_{j=0}^{r-1} \sum_{\theta \in \{0,1\}^j} \max\left(|\lambda_{\theta 0}|, |\lambda_{\theta 1}|\right) \operatorname{diam}(\Omega_\theta) =: D. \tag{3.8}$$

Since $|\mathcal{X}_1| = m$, it follows that

$$W_1(\mu_{\mathcal{X}_1}, \mu_{\mathcal{X}_2}) \leq \frac{D}{m}. \tag{3.9}$$

Combining Steps 1 and 2. Recall that step 1 transforms the input data $\mathcal{X}$ with $|\mathcal{X}| = n$ into $\mathcal{X}_1$ with $|\mathcal{X}_1| = m = n + \operatorname{sign}(\lambda) \cdot \lfloor |\lambda| \rfloor$ by adding or removing points, depending on the sign of λ.

Case 1: $\lambda \geq 0$. Here $\mathcal{X}_1$ is obtained from $\mathcal{X}$ by adding $\lfloor \lambda \rfloor$ points, so Lemma 3.17 gives

$$W_1(\mu_{\mathcal{X}}, \mu_{\mathcal{X}_1}) \leq \frac{\lambda}{m} \cdot \Delta_0.$$

Combining this with (3.9) by triangle inequality, we conclude that

$$W_1(\mu_{\mathcal{X}}, \mu_{\mathcal{X}_2}) \leq \frac{\lambda \Delta_0 + D}{m} \leq \frac{\lambda \Delta_0 + D}{n}.$$

Case 2: $\lambda < 0$. Here $\mathcal{X}_1$ is obtained from $\mathcal{X}$ by removing a set $\mathcal{X}_0$ of $n - m = \lfloor |\lambda| \rfloor$ points. Furthermore, by our analysis of step 2, $\mathcal{X}_2$ is obtained from $\mathcal{X}_1$ by moving points the total distance at most D. Therefore, $\mathcal{X}_2 \cup \mathcal{X}_0$ (as a multiset) is obtained from $\mathcal{X} = \mathcal{X}_1 \cup \mathcal{X}_0$ by moving points the total distance at most D, too. (The points in $\mathcal{X}_0$ remain unmoved.) Since $|\mathcal{X}| = n$, it follows that

$$W_1(\mu_{\mathcal{X}}, \mu_{\mathcal{X}_2 \cup \mathcal{X}_0}) \leq \frac{D}{n}.$$

Moreover, Lemma 3.17 gives

$$W_1(\mu_{\mathcal{X}_2}, \mu_{\mathcal{X}_2 \cup \mathcal{X}_0}) \leq \frac{|\mathcal{X}_0|}{|\mathcal{X}_2 \cup \mathcal{X}_0|} \cdot \operatorname{diam}(\Omega) \leq \frac{|\lambda| \, \Delta_0}{n}.$$

(Here we used that the multiset $\mathcal{X}_2 \cup \mathcal{X}_0$ has the same number of points as $\mathcal{X}$, which is n.) Combining the two bounds by triangle inequality, we obtain

$$W_1(\mu_{\mathcal{X}}, \mu_{\mathcal{X}_2}) \leq \frac{|\lambda| \, \Delta_0 + D}{n}. \tag{3.10}$$

In other words, this bound holds in both cases.

Analyzing Step 3. This step is the easiest to analyze: since $\mathcal{Y}$ is obtained from $\mathcal{X}_2$ by relocating the points are relocated within their cells, and the maximal diameter of the cells is δ, we have $W_1(\mu_{\mathcal{X}_2}, \mu_{\mathcal{Y}}) \leq \delta$. Combining this with (3.10) by triangle inequality, we conclude

that

$$W_1(\mu_{\mathcal{X}}, \mu_{\mathcal{Y}}) \leq \frac{|\lambda|\,\Delta_0 + D}{n} + \delta.$$

Taking expectation. Recall the definition of D from (3.8). We get

$$\mathbb{E}\, W_1(\mu_{\mathcal{X}}, \mu_{\mathcal{Y}}) \leq \frac{1}{n}\left[\mathbb{E}\left[|\lambda|\right]\Delta_0 + \sum_{j=0}^{r-1}\sum_{\theta\in\{0,1\}^j}\mathbb{E}\left[\max\left(|\lambda_{\theta 0}|\,,|\lambda_{\theta 1}|\right)\right]\operatorname{diam}(\Omega_\theta)\right] + \delta.$$

Since $\lambda \sim \mathrm{Lap}_{\mathbb{Z}}(\sigma_0)$, we have $\mathbb{E}\left[|\lambda|\right] \leq (\mathbb{E}(\lambda)^2)^{1/2} \leq \sqrt{2}\sigma_0$. Similarly, since $\lambda_{\theta 0}$ and $\lambda_{\theta 1}$ are independent $\mathrm{Lap}_{\mathbb{Z}}(\sigma_{j+1})$ random variables, $\mathbb{E}\left[\max\left(|\lambda_{\theta 0}|\,,|\lambda_{\theta 1}|\right)\right] \leq 2\sqrt{2}\sigma_{j+1}$. Substituting these estimates and rearranging the terms of the sum will complete the proof. □

Chapter 4

Data with Low-dimensional Structure

For the general synthetic data algorithm on $([0,1]^d, \|\cdot\|_\infty)$ we introduce in the last chapter, both PSMM and PMM has a factor $n^{-1/d}$ in the 1-Wasserstein error bound, according to Theorem 3.6 and Theorem 3.9. Such factor implies that the 1-Wasserstein error converges to 0 when the size of input data set is sufficiently large. However, this bound becomes meaningless when data is from a high-dimensional spaces, such as the scenario where $d = \Omega(\log n)$, because the 1-Wasserstein distance is always naturally bounded by 1 for the probability measures on $([0,1]^d, \|\cdot\|_\infty)$.

When the input data set has low-dimensional structure, such as lying on a d'-dimensional subspace inside the hypercube $[0,1]^d$, it is natural to expect a better bound of $n^{-1/d'}$ instead of $n^{-1/d}$. If such improvement is possible, we would overcome the curse of high dimensionality as we are capable of generating synthetic data in high-dimensional spaces. In this chapter, we will introduce an algorithm that captures the low-dimensional structure of the input data while applying general synthetic data algorithms.

In this section, we work with data in the Euclidean space $\mathbb{R}^d$ and $d' < d$. For convenience, the data matrix $\mathbf{X} = [X_1, \ldots, X_n] \in \mathbb{R}^{d\times n}$ also indicates the dataset $(X_1, \ldots, X_n)$. We use

$\mathbf{A}$ to denote a matrix and v, X as vectors. $\|\cdot\|_F$ denotes the Frobenius norm and $\|\cdot\|$ is the operator norm of a matrix. Two sequences a_n, b_n satisfies $a_n \lesssim b_n$ if $a_n \leq Cb_n$ for an absolute constant $C > 0$. For a matrix $\mathbf{V} \in \mathbb{R}^{d\times d'}$ with orthonormal column vectors, we also use $\mathbf{V}$ to indicate the subspace spanned by the column vectors of $\mathbf{V}$.

4.1 Outline of the main algorithm

From the input dataset $\mathbf{X}$, we aim to generate synthetic data $\mathbf{Y}$ with differential privacy and small error in 1-Wasserstein distance. The main algorithm Algorithm 5 can be summarized as in the following four steps:

1. Construct a private covariance matrix $\widehat{\mathbf{M}}$. The private covariance is constructed by adding a Laplacian random matrix to a centered covariance matrix $\mathbf{M}$ defined as

$$\mathbf{M} = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \overline{X})(X_i - \overline{X})^\top, \quad \text{where} \quad \overline{X} = \frac{1}{n}\sum_{i=1}^{n} X_i. \tag{4.1}$$

   This step is presented in Algorithm 6.

2. Find a d'-dimensional subspace $\widehat{\mathbf{V}}_{d'}$ by taking the top d' eigenvectors of $\widehat{\mathbf{M}}$. Then, project the data onto a linear subspace. The new data obtained in this way are inside a d'-dimensional ball. This step is summarized in Algorithm 7.

3. Generate a private measure in the d' dimensional ball centered at the origin by adapting methods in [53], where synthetic data generation algorithms were analyzed for data in the hypercube. This is summarized in Algorithms 9 and Algorithm 8.

4. Add a private mean vector to shift the dataset back to a private affine subspace. Given the transformations in earlier steps, some synthetic data points might lie outside the hypercube. We then metrically project them back to the domain of the hypercube.

Finally, we output the resulting dataset $\mathbf{Y}$. This is summarized in the last two parts of Algorithm 5.

The detailed steps of the algorithm is presented in Algorithm 5. And the following theorem provides the privacy and accuracy of Algorithm 5. The proof of the theorem is deferred to Section 4.4.

Theorem 4.1. *Let $\Omega = [0,1]^d$ equipped with ℓ^∞ metric and $\mathbf{X} = [X_1, \ldots, X_n] \in \Omega^n$ be a dataset. For any $2 \le d' \le d$, Algorithm 5 outputs an ε-differentially private synthetic dataset $\mathbf{Y} = [Y_1, \ldots, Y_m] \in \Omega^m$ for some $m \ge 1$ in polynomial time such that*

$$\mathbb{E}\, W_1(\mu_{\mathbf{X}}, \mu_{\mathbf{Y}}) \lesssim \sqrt{\sum_{i>d'} \sigma_i(\mathbf{M})} + \sqrt{\frac{d' d^{2.5}}{\varepsilon n}} + \sqrt{\frac{d}{d'}} (\varepsilon n)^{-1/d'}, \tag{4.2}$$

where $\sigma_i(\mathbf{M})$ is the i-th largest eigenvalue value of $\mathbf{M}$ in (4.1).

Note that m, the size of the synthetic dataset $\mathbf{Y}$, is not necessarily equal to n since the low-dimensional synthetic data subroutine in Algorithm 5 creates noisy counts. See Chapter 3 for more details.

Optimality There are three terms on the right-hand side of (4.20). The first term is the error from the rank-d' approximation of the covariance matrix $\mathbf{M}$. The second term is the accuracy loss for private PCA after the perturbation from a random Laplacian matrix. The optimality of this error term remains an open question. The third term is the accuracy loss when generating synthetic data in a d'-dimensional subspace. Notably, the factor $\sqrt{d/d'}$ is optimal. This can be seen by the fact that a d'-dimensional section of the cube can be $\sqrt{d/d'}$ times larger than the low-dimensional cube $[0,1]^{d'}$ (e.g., if it is positioned diagonally). Complementarily, [14] showed the optimality of the factor $(\varepsilon n)^{-1/d'}$ for generating d'-dimensional synthetic data in $[0,1]^{d'}$. Therefore, the third term in (4.20) is necessary and optimal.

Algorithm 5 Low-dimensional Synthetic Data

Input: True data matrix $\mathbf{X} = [X_1, \ldots, X_n]$, $X_i \in [0,1]^d$, privacy parameter ε.
(Private covariance matrix) Apply Algorithm 6 to $\mathbf{X}$ with privacy parameter $\varepsilon/3$ to obtain a private covariance matrix $\widehat{\mathbf{M}}$.
(Private linear projection) Let $\overline{X}_{\text{priv}}$ denote the private mean of the true dataset. Choose a target dimension d'. Apply Algorithm 7 with privacy parameter $\varepsilon/3$ to shift and project $\mathbf{X}$ onto a private d'-dimensional linear subspace.
(Low-dimensional synthetic data) Use subroutine in Section 4.3 to generate $\varepsilon/3$-DP synthetic data $\mathbf{X}'$ of size m depending on $d' = 2$ or $d' \geq 3$.
(Adding the private mean vector) Shift the data back by $X_i'' = X_i' + \overline{X}_{\text{priv}}$.
(Metric projection) Define $f : \mathbb{R} \to [0,1]$ such that

$$f(x) = \begin{cases} 0 & \text{if } x < 0; \\ x & \text{if } x \in [0,1]; \\ 1 & \text{if } x > 1. \end{cases}$$

Then, for $v \in \mathbb{R}^d$, we define $f(v)$ to be the result of applying f to each coordinate of v.
Output: Synthetic data $\mathbf{Y} = [f(X_1''), \ldots, f(X_m'')]$.

Improved accuracy When the original dataset $\mathbf{X}$ lies in an affine d'-dimensional subspace, it implies $\sigma_i(\mathbf{M}) = 0$ for $i > d'$ and $\mathbb{E}\, W_1(\mu_{\mathbf{X}}, \mu_{\mathbf{Y}}) \lesssim \sqrt{\frac{d'd^{2.5}}{\varepsilon n}} + \sqrt{\frac{d}{d'}}(\varepsilon n)^{-1/d'}$. This is an improvement from the accuracy rate $O((\varepsilon n)^{-1/d})$ for unstructured data in $[0,1]^d$ in [14, 53] when $d \leq n^{\alpha_n}$ and $d' \leq \min\{\frac{d}{2}, \frac{1}{\alpha_n}\}$ for $0 < \alpha_n \leq \frac{2}{7}$. For example, we can take α_n to be a constant in $(0, \frac{2}{7}]$ or $\alpha_n = \frac{1}{\log\log n}$. This improved rate overcomes the curse of high dimensionality.

Adaptive and private choices of d' The target dimension d' is a hyperparameter in Algorithm 5. One can choose the value of d' adaptively and privately based on singular values of the private covariance matrix $\widehat{\mathbf{M}}$ in Algorithm 6 such that

$$d' := \operatorname*{arg\,min}_{2 \leq k \leq d} \left(\sqrt{\sum_{i>d'} \sigma_i(\widehat{\mathbf{M}})} + \sqrt{\frac{d}{d'}}(\varepsilon n)^{-1/d'} \right).$$

Discussion on such choice of d' is referred to Section 4.5.

Low-dimensional representation of X. The generated synthetic dataset $\mathbf{Y}$ is close to a d'-dimensional subspace under the 1-Wasserstein distance, as shown in Proposition 4.8.

Running time The *private linear projection* step in Algorithm 5 has a running time $O(d^2 n)$ using the truncated SVD [70]. The *low-dimensional synthetic data* subroutine has a running time polynomial in n for $d' \geq 3$ and linear in n when $d' = 2$ [53]. Therefore, the overall running time for Algorithm 5 is linear in n, polynomial in d when $d' = 2$ and is poly(n, d) when $d' \geq 3$. Although sub-optimal in the dependence on d' for accuracy bounds, one can also run Algorithm 5 in linear time by choosing PMM (Algorithm 8) in the subroutine for all $d' \geq 2$.

4.2 Private linear projection

To reduce the dimension, we aim to find a d' dimensional private linear affine subspace and project data $\mathbf{X}$ onto it. Consider the $d \times n$ data matrix $\mathbf{X} = [X_1, \ldots, X_n]$, where $X_1, \ldots, X_n \in \mathbb{R}^d$. The rank of the covariance matrix $\frac{1}{n}\mathbf{X}\mathbf{X}^\top$ measures the dimension of the *linear subspace* spanned by $X_1, \ldots, X_n$. If we subtract the mean vector and consider the centered covariance matrix $\mathbf{M}$ in (4.1), then the rank of $\mathbf{M}$ indicates the dimension of the *affine linear subspace* that $\mathbf{X}$ lives in.

4.2.1 Private centered covariance matrix

To guarantee the privacy of $\mathbf{M}$, we add a symmetric Laplacian random matrix $\mathbf{A}$ to $\mathbf{M}$ to create a private Hermitian matrix $\widehat{\mathbf{M}}$ from Algorithm 6. The variance of entries in $\mathbf{A}$ is chosen such that the following privacy guarantee holds:

Proposition 4.2. *Algorithm 6 is ε-differentially private.*

Algorithm 6 Private Covariance Matrix

Input: Matrix $\mathbf{X} = [X_1, \ldots, X_n]$, privacy parameter ε.

(Computing the covariance matrix) Compute the mean $\overline{X} = \frac{1}{n}\sum_{i=1}^{n} X_i$ and the centered covariance matrix $\mathbf{M}$.

(Generating a Laplacian random matrix) Generate i.i.d random variables $\lambda_{ij} \sim \mathrm{Lap}(\sigma), i \le j$, where variance parameter $\sigma = \frac{3d^2}{\varepsilon n}$. Define a symmetric matrix $\mathbf{A}$ with

$$\mathbf{A}_{ij} = \mathbf{A}_{ji} = \begin{cases} \lambda_{ij} & \text{if } i < j; \\ 2\lambda_{ii} & \text{if } i = j, \end{cases}$$

Output: The noisy covariance matrix $\widehat{\mathbf{M}} = \mathbf{M} + \mathbf{A}$.

Proof. Before applying the definition of differential privacy, we compute the entries of $\mathbf{M}$ explicitly. One can easily check that

$$\mathbf{M} = \frac{1}{n}\sum_{k=1}^{n} X_k X_k^\top - \frac{1}{n(n-1)}\sum_{k\neq\ell} X_k X_\ell^\top. \tag{4.3}$$

Now, if there are neighboring datasets $\mathbf{X}$ and $\mathbf{X}'$, suppose $X_k = (X_k^{(1)}, \ldots, X_k^{(d)})^\top$ is a column vector in $\mathbf{X}$ and $X_k' = (X_k'^{(1)}, \ldots, X_k'^{(d)})^\top$ is a column vector in $\mathbf{X}'$, and all other column vectors are the same. Let $\mathbf{M}$ and $\mathbf{M}'$ be the covariance matrix of $\mathbf{X}$ and $\mathbf{X}'$, respectively. Then we consider the density function ratio for the output of Algorithm 6 with input $\mathbf{X}$ and $\mathbf{X}'$:

$$\begin{aligned}
\frac{\mathrm{den}_A(\widehat{\mathbf{M}} - \mathbf{M})}{\mathrm{den}_A(\widehat{\mathbf{M}} - \mathbf{M}')} &= \prod_{i<j} \frac{\mathrm{den}_{\lambda_{ij}}((\widehat{\mathbf{M}} - \mathbf{M})_{ij})}{\mathrm{den}_{\lambda_{ij}}((\widehat{\mathbf{M}} - \mathbf{M}')_{ij})} \prod_{i=j} \frac{\mathrm{den}_{2\lambda_{ij}}((\widehat{\mathbf{M}} - \mathbf{M})_{ij})}{\mathrm{den}_{2\lambda_{ij}}((\widehat{\mathbf{M}} - \mathbf{M}')_{ij})} \\
&= \prod_{i<j} \frac{\exp\left(-\frac{|(\widehat{\mathbf{M}}-\mathbf{M})_{ij}|}{\sigma}\right)}{\exp\left(-\frac{|(\widehat{\mathbf{M}}-\mathbf{M}')_{ij}|}{\sigma}\right)} \prod_{i} \frac{\exp\left(-\frac{|(\widehat{\mathbf{M}}-\mathbf{M})_{ii}|}{2\sigma}\right)}{\exp\left(-\frac{|(\widehat{\mathbf{M}}-\mathbf{M}')_{ii}|}{2\sigma}\right)} \\
&\le \exp\left(\sum_{i<j}\left|\mathbf{M}_{ij} - \mathbf{M}'_{ij}\right|/\sigma + \sum_{i}\left|\mathbf{M}_{ii} - \mathbf{M}'_{ii}\right|/(2\sigma)\right) \\
&= \exp\left(\frac{1}{2\sigma}\sum_{i,j}\left|\mathbf{M}_{ij} - \mathbf{M}'_{ij}\right|\right).
\end{aligned}$$

As the datasets differs on only one data X_k, consider all entry containing X_k in (4.3), we have

$$\begin{aligned}\left|\mathbf{M}_{ij} - \mathbf{M}'_{ij}\right| &\leq \frac{1}{n}\left|X_k^{(i)}X_k^{(j)} - X_k'^{(i)}X_k'^{(j)}\right| + \frac{1}{n(n-1)}\sum_{\ell\neq k}\left|X_k^{(i)} - X_k'^{(i)}\right| X_\ell^{(j)} \\ &\quad + \frac{1}{n(n-1)}\sum_{\ell\neq k} X_\ell^{(i)}\left|X_k^{(j)} - X_k'^{(j)}\right| \\ &\leq \frac{2}{n} + \frac{2}{n(n-1)}\cdot 2(n-1) \\ &= \frac{6}{n}.\end{aligned}$$

Therefore, substituting the result in the probability ratio implies

$$\frac{\mathrm{den}_A(\widehat{\mathbf{M}} - \mathbf{M})}{\mathrm{den}_A(\widehat{\mathbf{M}} - \mathbf{M}')} \leq \exp\left(\frac{1}{2\sigma}\cdot d^2 \cdot \frac{6}{n}\right) = \exp\left(\frac{3d^2}{\sigma n}\right),$$

and when $\sigma = \frac{3d^2}{\varepsilon n}$, Algorithm 6 is ε-differentially private. □

4.2.2 Noisy projection

The private covariance matrix $\widehat{\mathbf{M}}$ induces private subspaces spanned by eigenvectors of $\widehat{\mathbf{M}}$. We then perform a truncated SVD on $\widehat{\mathbf{M}}$ to find a private d'-dimensional subspace $\widehat{\mathbf{V}}_{d'}$ and project original data onto $\widehat{\mathbf{V}}_{d'}$. Here, the matrix $\widehat{\mathbf{V}}_{d'}$ also indicates the subspace generated by its orthonormal columns. The full steps are summarized in Algorithm 7.

Note that Algorithm 7 only guarantees privacy for the basis $\widehat{v}_1, \ldots, \widehat{v}_{d'}$ for each $\widehat{X}_i$, yet the coordinates of $\widehat{X}_i$ in terms of $\widehat{v}_1, \ldots, \widehat{v}_{d'}$ are *not private*. Synthetic data subroutines in the next stage will output synthetic data on the private subspace $\widehat{\mathbf{V}}_{d'}$ based on $\widehat{\mathbf{X}}$. The privacy analysis combines the two stages based on Lemma 2.4, and we state the results in Section 4.3.

Algorithm 7 Noisy Projection

Input: True data matrix $\mathbf{X} = [X_1, \ldots, X_n]$, $X_i \in [0,1]^d$, privacy parameters ε, the private covariance matrix $\widehat{\mathbf{M}}$ from Algorithm 6, and a target dimension d'.

(Singular value decomposition) Compute top d' orthonormal eigenvectors $\widehat{v}_1, \ldots, \widehat{v}_{d'}$ of $\widehat{\mathbf{M}}$ and denote $\widehat{\mathbf{V}}_{d'} = [\widehat{v}_1, \ldots, \widehat{v}_{d'}]$.

(Private centering) Compute $\overline{X} = \frac{1}{n}\sum_{i=1}^n X_i$. Let $\lambda \in \mathbb{R}^d$ be a random vector with i.i.d. components of $\mathrm{Lap}(d/(\varepsilon n))$. Shift each X_i to $X_i - (\overline{X} + \lambda)$ for $i \in [n]$.

(Projection) Project $\{X_i - (\overline{X} + \lambda)\}_{i=1}^n$ onto the linear subspace spanned by $\widehat{v}_1, \ldots, \widehat{v}_{d'}$. The projected data $\widehat{X}_i$ is given by $\widehat{X}_i = \sum_{j=1}^{d'} \langle X_i - (\overline{X} + \lambda), \widehat{v}_j \rangle \widehat{v}_j$.

Output: The data matrix after projection $\widehat{\mathbf{X}} = [\widehat{X}_1 \ldots \widehat{X}_n]$.

4.2.3 Accuracy guarantee for noisy projection

The data matrix $\widehat{\mathbf{X}}$ corresponds to an empirical measure $\mu_{\widehat{\mathbf{X}}}$ supported on the subspace $\widehat{\mathbf{V}}_d$. In this subsection, we characterize the 1-Wasserstein distance between the empirical measure $\mu_{\widehat{\mathbf{X}}}$ and the empirical measure of the centered dataset $\mathbf{X} - \overline{X}\mathbf{1}^\top$, where $\mathbf{1} \in \mathbb{R}^n$ is the all-1 vector. This problem can be formulated as the stability of a low-rank projection based on a covariance matrix with additive noise. We first provide the following useful deterministic lemma.

Lemma 4.3 (Stability of noisy projection)**.** *Let $\mathbf{X}$ be a $d \times n$ matrix and $\mathbf{A}$ be a $d \times d$ Hermitian matrix. Let $\mathbf{M} = \frac{1}{n}\mathbf{X}\mathbf{X}^\top$ with eigenvalues $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_d$. Let $\widehat{\mathbf{M}} = \frac{1}{n}\mathbf{X}\mathbf{X}^\top + \mathbf{A}$, $\widehat{\mathbf{V}}_{d'}$ be a $d \times d'$ matrix whose columns are the first d' orthonormal eigenvectors of $\widehat{\mathbf{M}}$, and $\mathbf{Y} = \widehat{\mathbf{V}}_{d'}\widehat{\mathbf{V}}_{d'}^\top\mathbf{X}$. Let $\mu_{\mathbf{X}}$ and $\mu_{\mathbf{Y}}$ be the empirical measures of column vectors of $\mathbf{X}$ and $\mathbf{Y}$, respectively. Then*

$$W_2^2(\mu_{\mathbf{X}}, \mu_{\mathbf{Y}}) \leq \frac{1}{n}\|\mathbf{X} - \mathbf{Y}\|_F^2 \leq \sum_{i>d'} \sigma_i + 2d'\|\mathbf{A}\|. \tag{4.4}$$

Proof. Let $\widehat{v}_1, \ldots, \widehat{v}_d$ be a set of orthonormal eigenvectors for $\widehat{\mathbf{M}}$ with the corresponding

eigenvalues $\widehat{\sigma}_1, \ldots, \widehat{\sigma}_d$. Define four matrices whose column vectors are eigenvectors:

$$\mathbf{V} = [v_1, \ldots, v_d], \qquad \widehat{\mathbf{V}} = [\widehat{v}_1, \ldots, \widehat{v}_d],$$
$$\mathbf{V}_{d'} = [v_1, \ldots, v_{d'}], \qquad \widehat{\mathbf{V}}_{d'} = [\widehat{v}_1, \ldots, \widehat{v}_{d'}].$$

By orthogonality, the following identities hold:

$$\sum_{i=1}^{d} \|v_i^\top \mathbf{X}\|_2^2 = \sum_{i=1}^{d} \|\widehat{v}_i^\top \mathbf{X}\|_2^2 = \|\mathbf{X}\|_F^2,$$
$$\sum_{i>d'} \|v_i^\top \mathbf{X}\|_2^2 = \|\mathbf{X} - \mathbf{V}_{d'}\mathbf{V}_{d'}^\top \mathbf{X}\|_F^2,$$
$$\sum_{i>d'} \|\widehat{v}_i^\top \mathbf{X}\|_2^2 = \|\mathbf{X} - \widehat{\mathbf{V}}_{d'}\widehat{\mathbf{V}}_{d'}^\top \mathbf{X}\|_F^2.$$

Separating the top d' eigenvectors from the rest, we obtain

$$\sum_{i\le d'} \|v_i^\top \mathbf{X}\|_2^2 + \|\mathbf{X} - \mathbf{V}_{d'}\mathbf{V}_{d'}^\top \mathbf{X}\|_F^2 = \sum_{i\le d'} \|\widehat{v}_i^\top \mathbf{X}\|_2^2 + \|\mathbf{X} - \widehat{\mathbf{V}}_{d'}\widehat{\mathbf{V}}_{d'}^\top \mathbf{X}\|_F^2.$$

Therefore

$$\begin{aligned}
\|\mathbf{X} - \widehat{\mathbf{V}}_{d'}\widehat{\mathbf{V}}_{d'}^\top \mathbf{X}\|_F^2 &= \sum_{i\le d'} \|v_i^\top \mathbf{X}\|_2^2 - \sum_{i\le d'} \|\widehat{v}_i^\top \mathbf{X}\|_2^2 + \|\mathbf{X} - \mathbf{V}_{d'}\mathbf{V}_{d'}^\top \mathbf{X}\|_F^2 \\
&= n\sum_{i\le d'} \sigma_i - n\sum_{i\le d'} \widehat{v}_i^\top \mathbf{M}\widehat{v}_i + n\sum_{i>d'} \sigma_i \\
&= n\sum_{i\le d'} \sigma_i - n\sum_{i\le d'} \widehat{v}_i^\top (\widehat{\mathbf{M}} - \mathbf{A})\widehat{v}_i + n\sum_{i>d'} \sigma_i \\
&= n\sum_{i\le d'} (\sigma_i - \widehat{\sigma}_i) + n\,\mathrm{tr}(\mathbf{A}\widehat{\mathbf{V}}_{d'}\widehat{\mathbf{V}}_{d'}^\top) + n\sum_{i>d'} \sigma_i. \qquad (4.5)
\end{aligned}$$

By Weyl's inequality, for $i \leq d'$,

$$|\sigma_i - \widehat{\sigma}_i| \leq \|\mathbf{A}\|. \tag{4.6}$$

By von Neumann's trace inequality,

$$\mathrm{tr}(A\widehat{\mathbf{V}}_{d'}\widehat{\mathbf{V}}_{d'}^\top) \leq \sum_{i=1}^{d'} \sigma_i(\mathbf{A}). \tag{4.7}$$

From (4.5), (4.6), and (4.7),

$$\frac{1}{n}\|\mathbf{X} - \widehat{\mathbf{V}}_{d'}\widehat{\mathbf{V}}_{d'}^\top\mathbf{X}\|_F^2 \leq \sum_{i>d'} \sigma_i + d'\|\mathbf{A}\| + \sum_{i=1}^{d'} \sigma_i(\mathbf{A}) \leq \sum_{i>d'} \sigma_i + 2d'\|\mathbf{A}\|.$$

Let Y_i be the i-th column of $\mathbf{Y}$. We have

$$W_2^2(\mu_{\mathbf{X}}, \mu_{\mathbf{Y}}) \leq \frac{1}{n}\sum_{i=1}^{n} \|X_i - Y_i\|_2^2 = \frac{1}{n}\|\mathbf{X} - \mathbf{Y}\|_F^2. \tag{4.8}$$

Therefore (4.4) holds. □

Note that inequality (4.4) holds without any spectral gap assumption on $\mathbf{M}$. Applying Davis-Kahan inequality would require $\sigma_{d'} - \sigma_{d'+1}$ to be large while Lemma 4.3 is applicable even when $\sigma_{d'} = \sigma_{d'+1}$. In the context of sample covariance matrices for random datasets, a related bound without a spectral gap condition is derived in [82, Proposition 2.2]. Furthermore, Lemma 4.3 bears a conceptual resemblance to [3, Theorem 5], which deals with low-rank matrix approximation under perturbation. With Lemma 4.3, we derive the following Wasserstein distance bounds between the centered dataset $\mathbf{X} - \overline{X}\mathbf{1}^\top$ and the dataset $\widehat{\mathbf{X}}$.

Proposition 4.4. *For input data* $\mathbf{X}$ *and output data* $\widehat{\mathbf{X}}$ *in Algorithm 7, let* $\mathbf{M}$ *be the covari-*

ance matrix defined in (4.1). *Assume* $n \geq 1/\varepsilon$. *Then for an absolute constant* $C > 0$,

$$\mathbb{E}\, W_1(\mu_{\mathbf{X}-\overline{X}\mathbf{1}^\top}, \mu_{\widehat{\mathbf{X}}}) \leq \left(\mathbb{E}\, W_2^2(\mu_{\mathbf{X}-\overline{X}\mathbf{1}^\top}, \mu_{\widehat{\mathbf{X}}})\right)^{1/2} \leq \sqrt{2\sum_{i>d'}\sigma_i(\mathbf{M})} + \sqrt{\frac{Cd'd^{2.5}}{\varepsilon n}}.$$

Proof. For the true covariance matrix $\mathbf{M}$, consider its SVD

$$\mathbf{M} = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \overline{X})(X_i - \overline{X})^\top = \sum_{j=1}^{d}\sigma_j v_j v_j^\top, \tag{4.9}$$

where $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_d$ are the singular values and $v_1 \dots v_d$ are corresponding orthonormal eigenvectors. Moreover, define two $d \times d'$ matrices

$$\mathbf{V}_{d'} = [v_1, \dots, v_{d'}], \qquad \widehat{\mathbf{V}}_{d'} = [\widehat{v}_1, \dots, \widehat{v}_{d'}].$$

Then the matrix $\widehat{\mathbf{V}}_{d'}\widehat{\mathbf{V}}_{d'}^\top$ is a projection onto the subspace spanned by the principal components $\widehat{v}_1, \dots, \widehat{v}_{d'}$.

In Algorithm 7, for any data X_i we first shift it to $X_i - \overline{X} - \lambda$ and then project it to $\widehat{\mathbf{V}}_{d'}\widehat{\mathbf{V}}_{d'}^\top(X_i - \overline{X} - \lambda)$. Therefore

$$\begin{aligned}\left\|X_i - \overline{X} - \widehat{\mathbf{V}}_{d'}\widehat{\mathbf{V}}_{d'}^\top(X_i - \overline{X} - \lambda)\right\|_\infty &\leq \left\|X_i - \overline{X} - \widehat{\mathbf{V}}_{d'}\widehat{\mathbf{V}}_{d'}^\top(X_i - \overline{X})\right\|_\infty + \left\|\widehat{\mathbf{V}}_{d'}\widehat{\mathbf{V}}_{d'}^\top\lambda\right\|_\infty \\ &\leq \left\|X_i - \overline{X} - \widehat{\mathbf{V}}_{d'}\widehat{\mathbf{V}}_{d'}^\top(X_i - \overline{X})\right\|_2 + \|\lambda\|_2 .\end{aligned}$$

Let Z_i denote $X_i - \overline{X}$ and $\mathbf{Z} = [Z_1, \dots, Z_n]$. Then

$$\frac{1}{n}\mathbf{Z}\mathbf{Z}^\top = \frac{n-1}{n}\mathbf{M}.$$

With Lemma 4.3, by definition of the Wasserstein distance, we have

$$W_2^2(\mu_{\mathbf{X}-\overline{X}\mathbf{1}^\top}, \mu_{\widehat{\mathbf{X}}}) \leq \frac{1}{n}\sum_{i=1}^{n}\left\|X_i - \overline{X} - \widehat{\mathbf{V}}_{d'}\widehat{\mathbf{V}}_{d'}^\top(X_i - \overline{X} - \lambda)\right\|_\infty^2 \tag{4.10}$$

$$\leq \frac{2}{n}\sum_{i=1}^{n}\left\|X_i - \overline{X} - \widehat{\mathbf{V}}_{d'}\widehat{\mathbf{V}}_{d'}^\top(X_i - \overline{X})\right\|_2^2 + 2\|\lambda\|_2^2 \tag{4.11}$$

$$= \frac{2}{n}\|\mathbf{Z} - \widehat{\mathbf{V}}_{d'}\widehat{\mathbf{V}}_{d'}^\top\mathbf{Z}\|_F^2 + 2\|\lambda\|_2^2 \tag{4.12}$$

$$\leq 2\sum_{i=d'}^{n}\sigma_i(\mathbf{M}) + 4d'\|\mathbf{A}\| + 2\|\lambda\|_2^2\,. \tag{4.13}$$

Since $\lambda = (\lambda_1, \ldots, \lambda_d)$ is a Laplacian random vector with i.i.d. $\mathrm{Lap}(1/(\varepsilon n))$ entries,

$$\mathbb{E}\|\lambda\|_2^2 = \sum_{j=1}^{d}\mathbb{E}\left|\lambda_j\right|^2 = \frac{2d}{\varepsilon^2 n^2}. \tag{4.14}$$

Furthermore, in Algorithm 6, A is a symmetric random matrix with independent Laplacian random variables on and above its diagonal. Thus, we have the tail bound for its norm [27, Theorem 1.1]

$$\mathbb{P}\left\{\|\mathbf{A}\| \geq \sigma(C\sqrt{d} + t)\right\} \leq C_0\exp(-C_1\min(t^2/4, t/2)). \tag{4.15}$$

And we can further compute the expectation bound for $\|\mathbf{A}\|$ from (4.15) with the choice of $\sigma = \frac{3d^2}{\varepsilon n}$,

$$\mathbb{E}\,\|\mathbf{A}\| \leq C\sigma\sqrt{d} + \int_0^\infty C_0\exp\left(-\,C_1\min\left(\frac{t^2}{4\sigma^2}, \frac{t}{2\sigma}\right)\right)\mathrm{d}t \lesssim \frac{d^{2.5}}{\varepsilon n}. \tag{4.16}$$

Combining the two bounds above and (4.13), as the 1-Wasserstein distance is bounded by

the 2-Wasserstein distance and inequality $\sqrt{x+y} \le \sqrt{x} + \sqrt{y}$ holds for all $x, y \ge 0$,

$$\begin{aligned}
\mathbb{E}\, W_1(\mu_{\mathbf{X}-\overline{X}\mathbf{1}^\top}, \mu_{\widehat{\mathbf{X}}}) &\le \left(\mathbb{E}\, W_2^2(\mu_{\mathbf{X}-\overline{X}\mathbf{1}^\top}, \mu_{\widehat{\mathbf{X}}})\right)^{1/2} \\
&\le \sqrt{2\sum_{i>d'} \sigma_i(\mathbf{M})} + \sqrt{4d'\,\mathbb{E}\|\mathbf{A}\|} + \sqrt{2\,\mathbb{E}\|\lambda\|_2^2} \\
&\le \sqrt{2\sum_{i>d'} \sigma_i(\mathbf{M})} + \sqrt{\frac{Cd'd^{2.5}}{\varepsilon n}},
\end{aligned}$$

where the last inequality holds under the assumption $\varepsilon n \ge 1$. □

4.3 Synthetic data subroutines

In the next stage of Algorithm 5, we construct synthetic data on the private subspace $\widehat{\mathbf{V}}_{d'}$ from the projected data. Since the original data X_i is in $[0,1]^d$, after Algorithm 7, we have

$$\left\|\widehat{X}_i\right\|_2 = \left\|X_i - \overline{X} - \lambda\right\|_2 \le \sqrt{d} + \left\|\overline{X} + \lambda\right\|_2 =: R \tag{4.17}$$

for any fixed $\lambda \in \mathbb{R}^d$. Therefore, the data after projection would lie in a d'-dimensional ball embedded in $\mathbb{R}^d$ with radius R, and the domain for the subroutine is

$$\Omega' = \{a_1\widehat{v}_1 + \cdots + a_{d'}\widehat{v}_{d'} \mid a_1^2 + \cdots + a_{d'}^2 \le R^2\},$$

where $\widehat{v}_1, \ldots, \widehat{v}_{d'}$ are the first d' private principal components in Algorithm 7.

Depending on whether $d' = 2$ or $d' \ge 3$, we apply two different algorithms from Chapter 3: private measure mechanism (PMM, Algorithm 4) and private signed measure mechanism (PSMM, Algorithm 1). We note that here $R = \sqrt{d} + \left\|\overline{X} + \lambda\right\|_2$ is a random variable depending on λ, but we will take R as a input parameter in the synthetic data subroutine and condition on R in this section.

4.3.1 $d' = 2$: private measure mechanism (PMM)

The synthetic data subroutine Algorithm 8 is adapted from the Private Measure Mechanism (PMM) in Algorithm 8. Recall that the PMM algorithm generates synthetic data in a hypercube by first partition the cube and then perturb the count in each sub-regions. It involves a binary hierarchical partition structure as defined in Definition 2.7.

To apply the binary hierarchical partition over the new region Ω' where projected data located, we first enlarge this ℓ_2-ball of radius R into a hypercube Ω_{PMM} of edge length $2R$ defined in Algorithm 8. Both the ℓ_2-ball Ω' and the larger hypercube Ω_{PMM} are inside the subspace $\widehat{\mathbf{V}}_{d'}$. The detailed description is as follows. The privacy and accuracy guarantees of Algorithm 8 are proved in the next proposition after stating the algorithm.

Algorithm 8 PMM subroutine after projection

Input: Privacy parameter ε, dataset $\widehat{\mathbf{X}} = (\widehat{X}_1, \ldots, \widehat{X}_n)$ in the region

$$\Omega' = \{a_1\widehat{v}_1 + \cdots + a_{d'}\widehat{v}_{d'} \mid a_1^2 + \cdots + a_{d'}^2 \le R\}.$$

(Binary partition) Let $r = \lceil \log_2(\varepsilon n) \rceil$ and $\sigma_j = \varepsilon^{-1} \cdot 2^{\frac{1}{2}(1-\frac{1}{d'})(r-j)}$. Enlarge the region Ω' into

$$\Omega_{\mathrm{PMM}} = \{a_1\widehat{v}_1 + \cdots + a_{d'}\widehat{v}_{d'} \mid a_i \in [-R, R], \forall i \in [d']\}.$$

Build a binary partition $\{\Omega_\theta\}_{\theta \in \{0,1\}^{\le r}}$ on Ω_{PMM}.
(Noisy count) For any θ, count the number of data in the region Ω_θ denoted by $n_\theta = \left|\widehat{\mathbf{X}} \cap \Omega_\theta\right|$, and let $n'_\theta = (n_\theta + \lambda_\theta)_+$, where λ_θ are independent integer Laplacian random variables with $\lambda \sim \mathrm{Lap}_{\mathbb{Z}}(\sigma_{|\theta|})$, where $|\theta|$ is the length of the vector θ and σ depends on ε as in (3.7).
(Consistency) Enforce consistency of $\{n'_\theta\}_{\theta \in \{0,1\}^{\le r}}$.
Output: Synthetic data $\mathbf{X}'$ generated by selecting n'_θ many data points arbitrarily (independently of $\widehat{\mathbf{X}}$) from Ω_θ for every $\theta \in \{0,1\}^r$.

Proposition 4.5. *The subroutine Algorithm 8 is ε-differentially private. Assume $n \ge 1/\varepsilon$. For any $d' \ge 2$, with the input as the projected data $\widehat{\mathbf{X}}$ and the range Ω' with radius R,*

Algorithm 8 has an accuracy bound

$$\mathbb{E}\, W_1(\mu_{\widehat{\mathbf{X}}}, \mu_{\mathbf{X}'}) \lesssim R(\varepsilon n)^{-1/d'},$$

where the expectation is taken with respect to the randomness of the synthetic data subroutine, conditioned on R.

Proof. The privacy guarantee follows from Theorem 3.8. For accuracy, note that the region Ω' is a subregion of a d'-dimensional ball. Algorithm 8 enlarges the region to a d'-dimensional hypercube with side length $2R$. By re-scaling the size of the hypercube and applying the result from 3.9, we obtain the accuracy bound. □

4.3.2 $d' \geq 3$: private signed measure mechanism (PSMM)

Recall that tge Private Signed Measure Mechanism (PSMM) also generates a synthetic dataset $\mathbf{Y}$ in a compact metric space Ω whose empirical measure $\mu_{\mathbf{Y}}$ is close to the empirical measure $\mu_{\mathbf{X}}$ of the original dataset $\mathbf{X}$ under the 1-Wasserstein distance. Compared to PMM, PSMM runs in polynomial time, and the partition only requires the diameters of the subregions are small enough. We have the following version of PSMM for our projected data inside the new low-dimensional domain Ω':

Here we note that, in the output step, such $\mathbf{X}'$ exist when the size of $\mathbf{X}'$ is large enough. The detailed discussion is presented in the beginning of Chapter 3 Section 3.1. As a result, we have the following result for the PSMM subroutine in this setting:

Proposition 4.6. *The subroutine Algorithm 9 is ε-differentially private. Assume $n \geq 1/\varepsilon$. When $d' \geq 3$, with the input as the projected data $\widehat{\mathbf{X}}$ and the range Ω' with radius R, the*

Algorithm 9 PSMM subroutine after projection

Input: Privacy parameter ε, dataset $\widehat{\mathbf{X}} = (\widehat{X}_1, \dots, \widehat{X}_n)$ in the region

$$\Omega' = \{a_1\widehat{v}_1 + \cdots + a_{d'}\widehat{v}_{d'} \mid a_1^2 + \cdots + a_{d'}^2 \le R^2\}.$$

(Integer lattice) Let $\delta = \frac{R}{\sqrt{d'}}(\varepsilon n)^{-1/d'}$. Find the lattice over the region:

$$L = \{a_1\widehat{v}_1 + \cdots + a_{d'}\widehat{v}_{d'} \mid a_1^2 + \cdots + a_{d'}^2 \le R^2, a_1, \dots, a_{d'} \in \delta\mathbb{Z}\}.$$

(Counting) For any $v = a_1\widehat{v}_1 + \cdots + a_{d'}\widehat{v}_{d'} \in L$, count the number

$$n_v = \left|\widehat{\mathbf{X}} \cap \{b_1\widehat{v}_1 + \cdots + b_{d'}\widehat{v}_{d'} \mid b_i \in [a_i, a_i + \delta)\}\right|.$$

(Adding noise) Define a synthetic signed measure ν such that for any $v \in L$,

$$\nu(\{v\}) = (n_v + \lambda_v)/n,$$

where $\lambda_v \sim \mathrm{Lap}_{\mathbb{Z}}(1/\varepsilon)$, $v \in L$ are i.i.d. random variables.
(Synthetic probability measure) Use linear programming in Algorithm 2 and find the closest probability measure $\widehat{\nu}$ to ν in d_{BL}-distance.
Output: Synthetic data $\mathbf{X}'$ containing copies of elements in L so that $\mu_{\mathbf{X}'}$ and $\widehat{\nu}$ are arbitrarily close.

algorithm has an accuracy bound

$$\mathbb{E}\, W_1(\mu_{\widehat{\mathbf{X}}}, \mu_{\mathbf{X}'}) \lesssim \frac{R}{\sqrt{d'}}(\varepsilon n)^{-1/d'}, \tag{4.18}$$

where the expectation is taken with respect to the randomness of the synthetic data subroutine, conditioned on R.

Proof. The proposition is a direct corollary to the result in [53]. The size of the scaled integer lattice $\delta\mathbb{Z}$ in the unit d-dimensional ball of radius R is bounded by $(\frac{CR}{\delta\sqrt{d'}})^d$ for an absolute constant $C > 0$ (see, for example, [43, Claim 2.9] and [13, Proposition 3.7]). Then, the number of subregions in Algorithm 9 is bounded by

$$|L| \le \left(\frac{R}{\sqrt{d'}} \cdot \frac{C}{\delta}\right)^{d'}.$$

By [53, Theorem 3.6], we have

$$\mathbb{E}\, W_1(\mu_{\widehat{\mathbf{X}}}, \mu_{\mathbf{X}'}) \le \delta + \frac{2}{\varepsilon n}\left(\frac{R}{\sqrt{d'}}\cdot\frac{C}{\delta}\right)^{d'}\cdot\frac{1}{d'}\left(\left(\frac{R}{\sqrt{d'}}\cdot\frac{C}{\delta}\right)^{d'}\right)^{-\frac{1}{d'}}.$$

Taking $\delta = \frac{CR}{\sqrt{d'}}(\varepsilon n)^{-\frac{1}{d'}}$ concludes the proof. □

Example 4.7 (PMM vs PSMM for $d' \ge 2$)**.** *For general $d' \ge 2$, PMM can still be applied, and the accuracy bound becomes $\mathbb{E}\, W_1(\mu_{\widehat{\mathbf{X}}}, \mu_{\mathbf{X}'}) \le CR(\varepsilon n)^{-1/d'}$. Compared to (4.18), the accuracy bound from PMM is weaker by a factor of $\sqrt{d'}$. However, as shown in [53], PMM has a running time linear in n and d, which is more computationally efficient than PSMM given in Algorithm 9 with running time polynomial in n, d.*

4.3.3 Re-centering and metric projection

After generating the private synthetic data, since we shift the data by its private mean before projection, we need to add another private mean vector back, which shifts the dataset $\widehat{\mathbf{X}}$ to a new private affine subspace close to the original dataset $\mathbf{X}$. The output data vectors in $\mathbf{X}''$ (defined in Algorithm 5) are not necessarily inside $[0,1]^d$. The subsequent metric projection enforces all synthetic data inside $[0,1]^d$. Importantly, this post-processing step does not have privacy costs.

After metric projection, dataset $\mathbf{Y}$ from the output of Algorithm 5 is close to an affine subspace, as shown in the next proposition. Notably, (4.19) shows that the metric projection step does not cause the largest accuracy loss among all subroutines.

Proposition 4.8 ($\mathbf{Y}$ is close to an affine subspace)**.** *The function $f : \mathbb{R}^d \to [0,1]^d$ in Algorithm 5 is the metric projection to $[0,1]^d$ with respect to $\|\cdot\|_\infty$, and the accuracy error*

for the metric projection step in Algorithm 5 is dominated by the error of the previous steps:

$$W_1(\mu_{\mathbf{Y}}, \mu_{\mathbf{X}''}) \leq W_1(\mu_{\mathbf{X}}, \mu_{\mathbf{X}''}), \tag{4.19}$$

where the dataset $\mathbf{X}''$ defined in Algorithm 5 is in a d'-dimensional affine subspace.

Proof. For the function f defined in Algorithm 5, we know $f(x)$ is the closest real number to x in the region $[0,1]$ for any $x \in \mathbb{R}$. Furthermore, if $v \in \mathbb{R}^d$ is a vector, then $f(v)$ is the closest vector to v in $[0,1]^d$ with respect to $\|\cdot\|_\infty$. Thus $f : \mathbb{R}^d \to [0,1]^d$ is indeed a metric projection to $[0,1]^d$.

We first assume that the synthetic data $\mathbf{X}''$ also has size n. Then for any column vector X_i'', we know that $Y_i = f(X_i'')$ is its closest vector in $[0,1]^d$ under the ℓ^∞ metric. For the data $X_1, X_2, \ldots, X_n$, suppose that the solution to the optimal transportation problem for $W_1(\mu_{\mathbf{X}}, \mu_{\mathbf{X}''})$ is to match $X_{\tau(i)}$ with X_i'', where τ is a permutation on $[n]$. Then

$$W_1(\mu_{\mathbf{Y}}, \mu_{\mathbf{X}''}) \leq \frac{1}{n}\sum_{i=1}^{n}\left\|Y_i - X_i''\right\|_\infty \leq \frac{1}{n}\sum_{i=1}^{n}\left\|X_{\tau(i)} - X_i''\right\|_\infty = W_1(\mu_{\mathbf{X}}, \mu_{\mathbf{X}''}).$$

In general, if the synthetic dataset has m data points and $m \neq n$, we can split the points and regard both the true dataset and synthetic dataset as of size mn, then it's easy to check that the inequality still holds. □

4.4 Privacy and accuracy of Algorithm 5

In this section, we summarize the privacy and accuracy guarantees of Algorithm 5. The privacy guarantee is proved by analyzing three parts of our algorithms: private mean, private linear subspace, and private data on an affine subspace.

Proposition 4.9 (Privacy)**.** *Algorithm 5 is ε-differentially private.*

Proof. We can decompose Algorithm 5 into the following steps:

1. $\mathcal{A}_1(\mathbf{X}) = \widehat{\mathbf{M}}$ computes the private covariance matrix with Algorithm 6.
2. $\mathcal{A}_2(\mathbf{X}) = \overline{X}_{\text{priv}} = \overline{X} + \lambda$ computes the private sample mean.
3. $\mathcal{A}_3(\mathbf{X}, y, \Sigma)$ for fixed y and Σ, is to project the shifted data $\{X_i - y\}_{i=1}^n$ to the first d' principal components of Σ and apply a certain differentially private subroutine (we choose y and Σ as the output of $\mathcal{A}_2$ and $\mathcal{A}_1$, respectively). This step outputs synthetic data $\mathbf{X}' = (X'_1, \ldots, X'_m)$ on a linear subspace.
4. $\mathcal{A}_4(\mathbf{X}', y)$ is to shift the dataset to $\{X'_i + y\}_{i=1}^m$ by a fixed vector y and applying the metric projection afterwards.

It suffices to show that the data before metric projection has already been differentially private. We will need to apply Lemma 2.4 several times.

With respect to the input $\mathbf{X}$ while fixing other input variables, we know that $\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3$ are all $\varepsilon/3$-differentially private. Therefore, by using Lemma 2.4 iteratively, the composition algorithm

$$\mathcal{A}_4(\mathcal{A}_3(\mathbf{X}, y, \mathcal{A}_1(\mathbf{X})), y), \quad \text{where } y = \overline{X}_{\text{priv}} = \mathcal{A}_2(\mathbf{X}),$$

satisfies ε-differential privacy. Hence Algorithm 5 is ε-differentially private. □

The next theorem combines errors from linear projection, synthetic data subroutine using PMM or PSMM, and the post-processing error from mean shift and metric projection.

Proposition 4.10 (Accuracy)**.** *For any given $2 \le d' \le d$ and $n \ge 1/\varepsilon$, the output data* $\mathbf{Y}$

from Algorithm 5 with the input data $\mathbf{X}$ *satisfies*

$$\mathbb{E}\, W_1(\mu_{\mathbf{X}}, \mu_{\mathbf{Y}}) \lesssim \sqrt{\sum_{i>d'} \sigma_i(\mathbf{M})} + \sqrt{\frac{d' d^{2.5}}{\varepsilon n}} + \sqrt{\frac{d}{d'}} (\varepsilon n)^{-1/d'}, \tag{4.20}$$

where $\mathbf{M}$ *denotes the covariance matrix in* (4.1).

Proof. In the case of $n < 1/\varepsilon$, we have $W_1(\mu_{\mathbf{X}}, \mu_{\mathbf{Y}}) \le 1 \le (\varepsilon n)^{-1/d'}$. The result is trivial. We assume $n \ge 1/\varepsilon$ in the rest of the proof.

Similar to privacy analysis, we will decompose the algorithm into several steps. Suppose that

1. $\mathbf{X} - (\overline{X} + \lambda)\mathbf{1}^\top$ denotes the shifted data $\{X_i - \overline{X} - \lambda\}_{i=1}^n$;
2. $\widehat{\mathbf{X}}$ is the data after projection to the private linear subspace;
3. $\mathbf{X}'$ is the output of the synthetic data subroutine in Section 4.3;
4. $\mathbf{X}'' = \mathbf{X}' + (\overline{X} + \lambda)\mathbf{1}^\top$ denotes the data shifted back;
5. $\mathcal{A}(\mathbf{X})$ is the data after metric projection, which is the output of the whole algorithm.

For the metric projection step, by Proposition 4.8, we have that

$$W_1(\mu_{\mathbf{X}}, \mu_{\mathcal{A}(\mathbf{X})}) \le W_1(\mu_{\mathbf{X}}, \mu_{\mathbf{X}''}) + W_1(\mu_{\mathbf{X}''}, \mu_{\mathcal{A}(\mathbf{X})}) \le 2W_1(\mu_{\mathbf{X}}, \mu_{\mathbf{X}''}). \tag{4.21}$$

Moreover, applying the triangle inequality of Wasserstein distance to the other steps of the

algorithm, we have

$$W_1(\mu_{\mathbf{X}}, \mu_{\mathbf{X}''}) = W_1(\mu_{\mathbf{X}-\overline{X}\mathbf{1}^\top}, \mu_{\mathbf{X}'+\lambda\mathbf{1}^\top}) \tag{4.22}$$

$$\leq W_1(\mu_{\mathbf{X}-\overline{X}\mathbf{1}^\top}, \mu_{\widehat{\mathbf{X}}}) + W_1(\mu_{\widehat{\mathbf{X}}}, \mu_{\mathbf{X}'}) + W_1(\mu_{\mathbf{X}'}, \mu_{\mathbf{X}'+\lambda}) \tag{4.23}$$

$$\leq W_1(\mu_{\mathbf{X}-\overline{X}\mathbf{1}^\top}, \mu_{\widehat{\mathbf{X}}}) + W_1(\mu_{\widehat{\mathbf{X}}}, \mu_{\mathbf{X}'}) + \|\lambda\|_\infty . \tag{4.24}$$

Note that $W_1(\mu_{\mathbf{X}-\overline{X}\mathbf{1}^\top}, \mu_{\widehat{\mathbf{X}}})$ is the projection error we bound in Theorem 4.4 with $n \geq 1/\varepsilon$, and $W_1(\mu_{\widehat{\mathbf{X}}}, \mu_{\mathbf{X}'})$ is treated in the accuracy analysis of subroutines in Section 4.3. Moreover, we have

$$\begin{aligned}
\mathbb{E}\, W_1(\mu_{\widehat{\mathbf{X}}}, \mu_{\mathbf{X}'}) &= \mathbb{E}_R\, \mathbb{E}_{\mathbf{X}'}\, W_1(\mu_{\widehat{\mathbf{X}}}, \mu_{\mathbf{X}'}) \\
&\leq \mathbb{E}_R\, \frac{CR}{\sqrt{d'}} (\varepsilon n)^{-1/d'} \\
&\leq \frac{C(2\sqrt{d} + \mathbb{E}\|\lambda\|_2)}{\sqrt{d'}} (\varepsilon n)^{-1/d'} \\
&\lesssim \sqrt{\frac{d}{d'}} (\varepsilon n)^{-1/d'}.
\end{aligned}$$

Here in the last step we use $\mathbb{E}\|\lambda\|_2 \leq \frac{C\sqrt{d}}{\varepsilon n}$ in (4.14). Since λ is a sub-exponential random vector, the following bound also holds for some absolute constant $C > 0$:

$$\mathbb{E}\big\|\lambda'\big\|_\infty \leq \frac{C \log d}{\varepsilon n}. \tag{4.25}$$

Hence

$$\mathbb{E}\, W_1(\mu_{\mathbf{X}}, \mu_{\mathcal{A}(\mathbf{X})}) \leq 2\, \mathbb{E}\, W_1(\mu_{\mathbf{X}}, \mu_{\mathbf{X}'+(\overline{X}+\lambda)\mathbf{1}^\top}) \tag{4.26}$$

$$\leq 2\, \mathbb{E}\, W_1(\mu_{\mathbf{X}-\overline{X}\mathbf{1}^\top}, \mu_{\widehat{\mathbf{X}}}) + 2\, \mathbb{E}\, W_1(\mu_{\widehat{\mathbf{X}}}, \mu_{\mathbf{X}'}) + 2\, \mathbb{E}\|\lambda\|_\infty \tag{4.27}$$

$$\leq 2\sqrt{2\sum_{i>d'} \sigma_i(\mathbf{M})} + 2\sqrt{\frac{Cd'd^{2.5}}{\varepsilon n}} + 2C\sqrt{\frac{d}{d'}} (\varepsilon n)^{-1/d'} + \frac{2C \log d}{\varepsilon n} \tag{4.28}$$

$$\lesssim \sqrt{\sum_{i>d'} \sigma_i(\mathbf{M})} + \sqrt{\frac{d}{d'}}(\varepsilon n)^{-1/d'} + \sqrt{\frac{d' d^{2.5}}{\varepsilon n}}, \tag{4.29}$$

where the first inequality is from (4.21), the second inequality is from (4.24), and the third inequality is due to Theorem 4.4, Proposition 4.5, and Proposition 4.6. □

4.5 Adaptive and private choice of d'

In our main Algorithm 5, d' is regarded as a fixed input hyper-parameter. In this section, we will show that it is possible to choose d' privately without sacrificing accuracy.

Lemma 4.11. *For* $\mathbf{M}$ *and* $\widehat{\mathbf{M}}$ *defined in Algorithm 6, there is*

$$\left| \sum_{i>d'} \sigma_i(\widehat{\mathbf{M}}) - \sum_{i>d'} \sigma_i(\mathbf{M}) \right| \lesssim \frac{(d-d')d^{2.5}}{\varepsilon n},$$

with probability at least $1 - C\exp(-c\sqrt{d})$.

Proof. By Weyl's inequality, $\left|\sigma_i(\widehat{\mathbf{M}}) - \sigma_i(\mathbf{M})\right| \leq \|\mathbf{A}\|$. Applying the (4.15) of the noise $\mathbf{A}$ implies the inequality in the lemma. □

Therefore, from Proposition 4.10, with probability at least $1 - C\exp(-c\sqrt{d})$, we have the following accuracy bound

$$\mathbb{E}\, W_1(\mu_{\mathbf{X}}, \mu_{\mathbf{Y}}) \lesssim \sqrt{\sum_{i>d'} \sigma_i(\mathbf{M})} + \sqrt{\frac{d' d^{2.5}}{\varepsilon n}} + \sqrt{\frac{d}{d'}}(\varepsilon n)^{-1/d'} \tag{4.30}$$

$$\lesssim \sqrt{\sum_{i>d'} \sigma_i(\widehat{\mathbf{M}}) + \frac{(d-d')d^{2.5}}{\varepsilon n}} + \sqrt{\frac{d' d^{2.5}}{\varepsilon n}} + \sqrt{\frac{d}{d'}}(\varepsilon n)^{-1/d'} \tag{4.31}$$

$$\lesssim \sqrt{\sum_{i>d'} \sigma_i(\widehat{\mathbf{M}})} + \sqrt{\frac{d}{d'}}(\varepsilon n)^{-1/d'} + \sqrt{\frac{d^{3.5}}{\varepsilon n}}. \tag{4.32}$$

Since the last term above is not related to d', we can choose

$$d' \coloneqq \arg\min_{2\le k\le d} \left(\sqrt{\sum_{i>k} \sigma_i(\widehat{\mathbf{M}})} + \sqrt{\frac{d}{k}} (\varepsilon n)^{-1/k} \right)$$

After computing the private covariance matrix in the first step. The privacy of the choice of d' is guaranteed as we only use the private covariance matrix $\mathbf{M}$.

4.6 Near-optimal accuracy bound with additional assumptions when $d' = 1$

Our Proposition 4.10 is not applicable to the case $d' = 1$ because the projection error in Theorem 4.4 only has bound $O((\varepsilon n)^{-\frac{1}{2}})$, which does not match with the optimal synthetic data accuracy bound in [14, 53]. We are able to improve the accuracy bound with an additional dependence on $\sigma_1(\mathbf{M})$ as follows:

Theorem 4.12. *When $d' = 1$, consider Algorithm 5 with input data $\mathbf{X}$, output data $\mathbf{Y}$, and the subroutine PMM in Algorithm 8. Let $\mathbf{M}$ be the covariance matrix defines as (4.1). Assume $\sigma_1(\mathbf{M}) > 0$ and $n \ge 1/\varepsilon$, then*

$$\mathbb{E}\, W_1(\mu_{\mathbf{X}}, \mu_{\mathbf{Y}}) \lesssim \sqrt{\sum_{i>1} \sigma_i(\mathbf{M})} + \frac{d^3}{\sqrt{\sigma_1(\mathbf{M})}\varepsilon n} + \frac{\sqrt{d}\log^2(\varepsilon n)}{\varepsilon n}.$$

We start with the following lemma based on the Davis-Kahan theorem [99].

Lemma 4.13. *Let $\mathbf{X}$ be a $d\times n$ matrix and $\mathbf{A}$ be an $d\times d$ Hermitian matrix. Let $\mathbf{M} = \frac{1}{n}\mathbf{X}\mathbf{X}^\top$, with the SVD*

$$\mathbf{M} = \sum_{j=1}^{d} \sigma_j v_j v_j^\top,$$

where $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_d$ *are the singular values of* $\mathbf{M}$ *and* $v_1, \ldots, v_d$ *are corresponding orthonormal eigenvectors. Let* $\widehat{\mathbf{M}} = \frac{1}{n}\mathbf{X}\mathbf{X}^\top + \mathbf{A}$ *with orthonormal eigenvectors* $\widehat{v}_1, \ldots, \widehat{v}_d$*, where* $\widehat{v}_1$ *corresponds to the top singular value of* $\widehat{\mathbf{M}}$*. When there exists a spectral gap* $\sigma_1 - \sigma_2 = \delta > 0$*, we have*

$$\frac{1}{n}\|\mathbf{X} - \widehat{v}_1\widehat{v}_1^\top \mathbf{X}\|_F^2 \leq 2\sum_{i>1}\sigma_i + \frac{8}{n\delta^2}\|\mathbf{A}\|^2\|\mathbf{X}\|_F^2 .$$

Proof. We have that

$$\begin{aligned}
\frac{1}{n}\|\mathbf{X} - \widehat{v}_1\widehat{v}_1^\top \mathbf{X}\|_F^2 &= \frac{1}{n}\|\mathbf{X} - v_1 v_1^\top \mathbf{X} + v_1 v_1^\top \mathbf{X} - \widehat{v}_1\widehat{v}_1^\top \mathbf{X}\|_F^2 \\
&\leq \frac{2}{n}\left(\|\mathbf{X} - v_1 v_1^\top \mathbf{X}\|_F^2 + \|v_1 v_1^\top \mathbf{X} - \widehat{v}_1\widehat{v}_1^\top \mathbf{X}\|_F^2\right) \\
&= 2\sum_{i>1}\sigma_i + \frac{2}{n}\left\|\left(v_1 v_1^\top - \widehat{v}_1\widehat{v}_1^\top\right)\mathbf{X}\right\|_F^2 \\
&\leq 2\sum_{i>1}\sigma_i + \frac{2}{n}\left\|v_1 v_1^\top - \widehat{v}_1\widehat{v}_1^\top\right\|^2\|\mathbf{X}\|_F^2 . \qquad (4.33)
\end{aligned}$$

To bound the operator norm distance between the two projections, we will need the Davis-Kahan Theorem in the perturbation theory. For the angle $\Theta(v_1, \widehat{v}_1)$ between the vectors v_1 and $\widehat{v}_1$, applying [99, Corollary 1], we have

$$\left\|v_1 v_1^\top - \widehat{v}_1\widehat{v}_1^\top\right\| = \sin\Theta(v_1, \widehat{v}_1) \leq \frac{2\|\mathbf{M} - \widehat{\mathbf{M}}\|}{\sigma_1 - \sigma_2} \leq \frac{2\|\mathbf{A}\|}{\delta}.$$

Therefore, when the spectral gap exists ($\delta > 0$),

$$\frac{1}{n}\|\mathbf{X} - \widehat{v}_1\widehat{v}_1^\top \mathbf{X}\|_F^2 \leq 2\sum_{i>1}\sigma_i + \frac{8}{n\delta^2}\|\mathbf{A}\|^2\|\mathbf{X}\|_F^2 .$$

This finishes the proof. □

Compared to Lemma 4.3, with the extra spectral gap assumption, the dependence on $\mathbf{A}$ in

the upper bound changes from $\|\mathbf{A}\|$ to $\|\mathbf{A}\|^2$. A similar phenomenon, called *global and local bounds*, was observed in [82, Proposition 2.2]. With Lemma 4.13, we are able to improve the accuracy rate for the noisy projection step as follows.

Proposition 4.14. *Let $\sigma_1 \geq \cdots \geq \sigma_d \geq 0$ be the singular values of $\mathbf{M}$ defined in (4.9). When $d' = 1$, assume that $\sigma_1 > 0$ and $n \geq 1/\varepsilon$. For the output $\widehat{\mathbf{X}}$ in Algorithm 7, we have*

$$\mathbb{E}\, W_1(\mu_{\mathbf{X}-\overline{X}\mathbf{1}^\top}, \mu_{\widehat{\mathbf{X}}}) \leq \left(\mathbb{E}\, W_2^2(\mu_{\mathbf{X}-\overline{X}\mathbf{1}^\top}, \mu_{\widehat{\mathbf{X}}})\right)^{1/2} \lesssim \sqrt{\sum_{i>1}\sigma_i} + \frac{d^3}{\sqrt{\sigma_1}\varepsilon n},$$

Proof. Similar to the proof of Theorem 4.4, we can define $Z_i = X_i - \overline{X}$ and deduce that

$$\begin{aligned}
&\frac{1}{n}\mathbf{Z}\mathbf{Z}^\top = \frac{n-1}{n}\mathbf{M},\\
&\frac{1}{n}\|\mathbf{Z}\|_F^2 = \frac{n-1}{n}\operatorname{tr}(\mathbf{M}),
\end{aligned}$$

and

$$W_2^2(\mu_{\mathbf{X}-\overline{X}\mathbf{1}^\top}, \mu_{\widehat{\mathbf{X}}}) = \frac{2}{n}\|\mathbf{Z} - \widehat{v}_1\widehat{v}_1^\top\mathbf{Z}\|_F^2 + 2\|\lambda\|_2^2\,.$$

By the inequality $\sqrt{x+y} \leq \sqrt{x} + \sqrt{y}$ for $x, y \geq 0$,

$$\mathbb{E}\, W_1(\mu_{\mathbf{X}-\overline{X}\mathbf{1}^\top}, \mu_{\widehat{\mathbf{X}}}) \leq \mathbb{E}\left[\frac{2}{n}\|\mathbf{Z} - \widehat{v}_1\widehat{v}_1^\top\mathbf{Z}\|_F^2\right]^{1/2} + \sqrt{2}\,\mathbb{E}\|\lambda\|_2\,.$$

Let $\delta = \sigma_1 - \sigma_2$. Next, we will discuss two cases for the value of δ.

Case 1: When $\delta = \sigma_1 - \sigma_2 \leq \frac{1}{2}\sigma_1$, we have $\sigma_1 \leq 2\sigma_2$ and

$$\operatorname{tr}(\mathbf{M}) = \sigma_1 + \cdots + \sigma_d \leq 3\sum_{i>1}\sigma_i.$$

As any projection map has spectral norm 1, we have $\left\|v_1v_1^\top - \widehat{v}_1\widehat{v}_1^\top\right\| \leq 2$. Applying (4.33),

we have

$$\begin{aligned}\frac{1}{n}\|\mathbf{Z}-\widehat{v}_1\widehat{v}_1^\top\mathbf{Z}\|_F^2 &\le 2\sum_{i>1}\sigma_i+\frac{2}{n}\left\|v_1v_1^\top-\widehat{v}_1\widehat{v}_1^\top\right\|^2\|\mathbf{Z}\|_F^2\\ &\le 2\sum_{i>1}\sigma_i+\frac{8}{n}\|\mathbf{Z}\|_F^2\\ &\le 2\sum_{i>1}\sigma_i+8\,\mathrm{tr}(\mathbf{M})\le 26\sum_{i>1}\sigma_i.\end{aligned}$$

Hence

$$\mathbb{E}\,W_1(\mu_{\mathbf{X}-\overline{X}\mathbf{1}^\top},\mu_{\widehat{\mathbf{X}}})\lesssim\sqrt{\sum_{i>1}\sigma_i}+\mathbb{E}\|\lambda\|_2\lesssim\sqrt{\sum_{i>1}\sigma_i}+\frac{\sqrt{d}}{\varepsilon n}. \tag{4.34}$$

Case 2: When $\delta\ge\frac{1}{2}\sigma_1$, we have

$$\mathrm{tr}(\mathbf{M})\le d\sigma_1\le\frac{4d\delta^2}{\sigma_1}.$$

For any fixed δ, by Lemma 4.13,

$$\begin{aligned}\frac{1}{n}\|\mathbf{Z}-\widehat{v}_1\widehat{v}_1^\top\mathbf{Z}\|_F^2 &\le 2\sum_{i>1}\sigma_i+\frac{8}{n\delta^2}\|\mathbf{A}\|^2\|\mathbf{Z}\|_F^2\\ &\le 2\sum_{i>1}\sigma_i+\frac{8}{\delta^2}\|\mathbf{A}\|^2\,\mathrm{tr}(\mathbf{M})\\ &\le 2\sum_{i>1}\sigma_i+\frac{32d}{\sigma_1}\|\mathbf{A}\|^2.\end{aligned}$$

So we have the Wasserstein distance bound

$$\mathbb{E}\,W_1(\mu_{\mathbf{X}-\overline{X}\mathbf{1}^\top},\mu_{\widehat{\mathbf{X}}})\le\sqrt{2\sum_{i>1}\sigma_i}+\sqrt{\frac{32d}{\sigma_1}}\,\mathbb{E}\|\mathbf{A}\|+\sqrt{2}\,\mathbb{E}\|\lambda\|_2 \tag{4.35}$$

$$\le\sqrt{2\sum_{i>1}\sigma_i}+\sqrt{\frac{32d}{\sigma_1}}\frac{d^{2.5}}{\varepsilon n}+\frac{\sqrt{2d}}{\varepsilon n} \tag{4.36}$$

$$\leq \sqrt{2\sum_{i>1}\sigma_i} + \frac{Cd^3}{\sqrt{\sigma_1}\varepsilon n}. \tag{4.37}$$

From (4.9),

$$\sigma_1 = \|M\| \leq \|M\|_F \leq \frac{n}{n-1}d \leq 2d.$$

Combining the two cases (4.34) and (4.37), we deduce the result. □

Proof of Theorem 4.12. Following the steps in the proof of Theorem 4.4, we obtain

$$\begin{aligned}
\mathbb{E}\, W_1(\mu_{\mathbf{X}}, \mu_{\mathcal{A}(\mathbf{X})}) &\leq 2\,\mathbb{E}\, W_1(\mu_{\mathbf{X}}, \mu_{\mathbf{X}'+(\overline{X}+\lambda')\mathbf{1}^\top}) \\
&\leq 2\,\mathbb{E}\, W_1(\mu_{\mathbf{X}-\overline{X}\mathbf{1}^\top}, \mu_{\widehat{\mathbf{X}}}) + 2\,\mathbb{E}\, W_1(\mu_{\widehat{\mathbf{X}}}, \mu_{\mathbf{X}'}) + 2\,\mathbb{E}\big\|\lambda'\big\|_\infty \\
&\lesssim \sqrt{\sum_{i>1}\sigma_i} + \frac{d'd^3}{\sqrt{\sigma_1}\varepsilon n} + \frac{\sqrt{d}\log^2(\varepsilon n)}{\varepsilon n} + \frac{2C\log d}{\varepsilon n} \\
&\lesssim \sqrt{\sum_{i>1}\sigma_i} + \frac{d'd^3}{\sqrt{\sigma_1}\varepsilon n} + \frac{\sqrt{d}\log^2(\varepsilon n)}{\varepsilon n},
\end{aligned}$$

where for the second inequality, we apply the bound from [53, Theorem 1.1] for the second term, and we use (4.25) for the third term. □

4.7 Conclusion

In this chapter, we provide a DP algorithm to generate synthetic data, which closely approximates the true data in the hypercube $[0,1]^d$ under 1-Wasserstein distance. Moreover, when the true data lies in a d'-dimensional affine subspace, we improve the accuracy guarantees in [53] and circumvents the curse of dimensionality by generating a synthetic dataset close to the affine subspace.

It remains open to determine the optimal dependence on d in the accuracy bound in Proposition 4.10 and whether the third term in (4.20) is needed. Our analysis of private PCA works without using the classical Davis-Kahan inequality that requires a spectral gap on the dataset. However, to approximate a dataset close to a line ($d' = 1$), additional assumptions are needed in our analysis to achieve the near-optimal accuracy rate, see Section 4.6. It is an interesting problem to achieve an optimal rate without the dependence on $\sigma_1(\mathbf{M})$ when $d' = 1$.

Our Algorithm 5 only outputs synthetic data with a low-dimensional linear structure, and its analysis heavily relies on linear algebra tools. For original datasets from a d'-dimensional linear subspace, we improve the accuracy rate from $(\varepsilon n)^{-1/(d'+1)}$ in [33] to $(\varepsilon n)^{-1/d'}$. It is also interesting to provide algorithms with optimal accuracy rates for datasets from general low-dimensional manifolds beyond the linear setting.

Chapter 5

Online Synthetic Data Generation

The online model, which is also referred as continual releasing model, describes a case where data is coming sequentially as $(X_t)_{t=1}^{\infty}$, and by time t only the first t data $(X_s)_{s=1}^{t}$ are accessible. More details under this setting can be found in Chapter 2, Section 2.4.2.

In this chapter, the size of the dataset keeps changing. For simplicity, we focus on the case where each time t there is exactly one data coming unless noted particularly, and at time t we have exactly t many data points.

As a warm-up, we first explain the complexity of the online case with two failing approaches by using the algorithms from Chapter 3:

1. To generate synthetic data based on the newly received data points. However, the error from Chapter 3 implies an error bound of $n^{-1/d}$ where n is the size of the updated data and it might be too small.

2. To generate synthetic data by combining old data and new data. However, if we ensure ε-differential privacy each time, as the earliest data is used multiple times, by Lemma 2.3, the total algorithm would be only (εt)-differentially private.

We consider the problem of generating private synthetic data under the continual release model beyond the Boolean data setting considered in [37, 22]. The data stream comes from the hypercube $[0,1]^d$ equipped with the ℓ_∞-norm, and our goal is to efficiently generate private synthetic data in an online fashion while maintaining a near-optimal utility bound under the Wasserstein distance.

Our main result is given in the next theorem.

Theorem 5.1 (Online private synthetic data)**.** *For any constant $\varepsilon > 0$, there is an ε-differentially private algorithm such that, for any data stream $X_1, \ldots, X_t, \cdots \in [0,1]^d$, at any time t, it transforms the first t points $\mathcal{X}_t = \{X_1, \ldots, X_t\}$ into t points $\mathcal{Y}_t \subset [0,1]^d$, with the following accuracy bound: there exists a constant C_ε depending only on ε such that for $t \geq C_\varepsilon$,*

$$\mathbb{E}W_1(\mu_{\mathcal{X}_t}, \mu_{\mathcal{Y}_t}) \lesssim \begin{cases} \log(t)(\varepsilon t)^{-\frac{1}{d}}, & d \geq 2, \\ \log^3(\varepsilon t)\log^{1.5}(t)(\varepsilon t)^{-1}, & d = 1, \end{cases} \tag{5.1}$$

where $W_1(\mu_{\mathcal{X}_t}, \mu_{\mathcal{Y}_t})$ is the 1-Wasserstein distance between two empirical measures $\mu_{\mathcal{X}_t}, \mu_{\mathcal{Y}_t}$ of $\mathcal{X}_t$ and $\mathcal{Y}_t$, respectively.

Example 5.2. *From the proof of Theorem 5.1 in Sections 5.4 and 5.5, we can choose*

$$C_\varepsilon = \begin{cases} \exp(\log^2(1/\varepsilon + 1)) + e/\varepsilon, & d \geq 2, \\ e/\varepsilon & d = 1. \end{cases}$$

Our algorithm is computationally efficient. To obtain synthetic datasets $\mathcal{Y}_t$ at time t, the time complexity is $O(dt + \varepsilon t \log t)$; see Section 5.6 for details.

The utility guarantee in Theorem 5.1 is optimal up to a $\log t$ factor for $d \geq 2$ and polylog(t) for $d = 1$. Compared to offline synthetic data tasks, generating online private synthetic

data is much more challenging, especially with an infinite time horizon. For offline private synthetic data on $[0,1]^d$, $d \geq 2$, [53] proposed an algorithm with utility bound $O(n^{-1/d})$, which matches the minimax lower bound proved in [14].

We prove Theorem 5.1 by analyzing our main Algorithm 12 for $d \geq 2$ and $d = 1$, respectively. We develop an online hierarchical partition procedure to divide the domain $[0,1]^d$ into disjoint sub-regions with decreasing diameters as time increases and then apply online private counting subroutines to count the number of data points in each subregion. After the online private counting step, we create synthetic data following the Consistency and Output steps described in Algorithm 12.

A key ingredient in our work is the development of a special *Inhomogeneous Sparse Counting Algorithm* (Algorithm 11) for the online private count of data points in each subregion, which has different privacy budgets for different time intervals. Such dynamic assignments are motivated by the selection of optimal privacy budgets based on the dynamic hierarchical partition. We apply the new counting algorithm with carefully designed privacy parameters and starting times for each subregion based on the hierarchical structure of the online partition.

The concept of counting sparse data also plays an important role. Intuitively, when inputs $X_1, \ldots, X_t$ are uniformly distributed in $[0,1]^d$, the online count of a newly created sub-region corresponds to a sum of a sparser Boolean data stream as the diameter of the sub-region decreases. In fact, the uniformly distributed data represents the worst-case configuration of the true dataset in the minimax lower bound proof in [14], corresponding to a sparse Boolean data stream for each subregion with a small diameter. We make use of the sparsity to obtain a near-optimal accuracy bound.

5.1 Dynamic Partition

We follow the definition of binary hierarchical partition as described in Definition 2.7 and consider the partition $\{\Omega_\theta \mid \theta \in \{0,1\}^{\leq r}\}$ Given a true data stream $(X_1, \ldots, X_t) \in \Omega^t$, the true count $n_\theta^{(t)}$ denotes the number of data points in the region Ω_θ at time t, i.e.,

$$n_\theta^{(t)} := \left| \left\{ i \in [t] : \ X_i \in \Omega_\theta \right\} \right|.$$

However, now the data points accumulates while t grows. At the beginning there are only a few data points and it is not necessary to deeply partition Ω; when t is large the size of true dataset is also large, and we need to partition the domain Ω into finer subregions to ensure accuracy. Therefore, a dynamically refining partition structure is needed under the online setting.

We can represent a binary hierarchical partition of Ω in a binary tree of depth r, where the root is labeled Ω and the j-th level of the tree $\mathcal{T}$ encodes the subsets Ω_θ for θ at level j. As new data arrives, we refine the binary partition over time and update the true count $n_\theta^{(t)}$ in each subregion at time t. As we continue refining the partition of Ω, the binary tree $\mathcal{T}$ expands in the order of a breadth-first search, and the online synthetic data we release will depend on a noisy count $N_\theta^{(t)}$ of data points in each region Ω_θ at time t.

5.2 Online Counting algorithms

As a simplest problem in the private online algorithm, we will first study the *private online counting mechanisms*: for the input data $(X_t)_{t=1}^T$ where $T \in \mathbb{N}_+ \cup \{\infty\}$ and each $X_t \in \{0,1\}$, we aim to output $Y_t = \sum_{i=1}^t X_i$ with differential privacy at time t.

5.2.1 Binary Mechanism

The *Binary Mechanism* is proposed and discussed in [37, 22]. Compared to the naive way of protecting ε/T-differential privacy at each time (due to Lemma 2.4), the core of the algorithm is to adding noise in a more elegant way which makes less influence to the accuracy: first partitioning the timeline binarily and then adding Laplacian noises in each node.

The details of the algorithm is referred to [22], where the authors fully discuss different cases of the private online counting with finite time horizon $T < \infty$ and infinite time horizon $T = \infty$. As a result, we present the results as follows.

Lemma 5.3 (Corollary 4.8 in [22])**.** *The Binary Mechanism is ε-differentially private for an infinite time horizon. And for any time $t \in [0, T]$ and $\beta > 0$, with probability at least $1 - \beta$, the counting error at time t is bounded by $\frac{C}{\varepsilon} \cdot \log^{1.5} T \cdot \log \frac{1}{\beta}$.*

From the probability bound of the error, we can also deduce an bound in expectation.

Lemma 5.4. *The Binary Mechanism is ε-differentially private for a finite time horizon T. And for any time $t \in [0, T]$, let error_t denote the error at time t between the true count $\sum_{i=1}^{t} X_i$ and the output at time t, we have*

$$\mathbb{E} \operatorname{error}_t \leq \frac{C}{\varepsilon} \log^{1.5} T. \tag{5.2}$$

Proof. Lemma 5.4 is a modified version of the following lemma, with the utility bound in expectation.

Although such error bound is in probability, we can easily transform it into a similar expectation bound. In fact, let error_t denote the error at time t between the true count $\sum_{i=1}^{t} X_i$ and the output at time t, we have

$$\mathbb{E} \operatorname{error}_t = \int_0^\infty \mathbb{P} \left\{ \operatorname{error}_t > u \right\} \mathrm{d}u.$$

After a change of variable $u = \frac{C}{\varepsilon} \cdot \log^{1.5} T \cdot \log \frac{1}{\beta}$ or $\beta = \exp\big(-\frac{\varepsilon u}{C \log^{1.5} T}\big)$, we can compute

$$\begin{aligned}
\mathbb{E}\,\mathrm{error}_t &= \int_0^\infty \mathbb{P}\left\{\mathrm{error}_t > u\right\} \mathrm{d}u \\
&= \int_0^1 \mathbb{P}\left\{\mathrm{error}_t > \frac{C}{\varepsilon} \cdot \log^{1.5} t \cdot \log \frac{1}{\beta}\right\} \frac{C \log^{1.5} T}{\beta\varepsilon} \mathrm{d}\beta \\
&\leq \int_0^1 \beta \cdot \frac{C \log^{1.5} T}{\beta\varepsilon} \mathrm{d}\beta \\
&= \frac{C}{\varepsilon} \log^{1.5} T.
\end{aligned}$$

Therefore, we have the expectation error bound

$$\mathbb{E}\,\mathrm{error}_t \leq \frac{C}{\varepsilon} \log^{1.5} T.$$

□

5.2.2 Sparse counting algorithm

With the Binary Mechanism, we solve the private online counting algorithm with a poly-log error bound in expectation. However, as [37] showed in the following lemma, such error is inevitable.

Lemma 5.5 (Theorem 4.2 in [37])**.** *Any differentially private event-level algorithm for counting over T rounds must have error $\Omega(\log T)$ (even with $\varepsilon = 1$).*

Nonetheless, there is still an gap between $\log^{1.5} T$ in Lemma 5.4 and the $\Omega(\log T)$ in the lower bound. In this section, we will introduce an algorithm from [39] to improve the counting error in private online counting problem with the help of the sparse structure inside the dataset.

The sparse counting algorithm with finite time horizon T in [39] is ε-differentially private with

an optimal accuracy error $O(\log T)$ when the data stream is sparse. Its idea is to partition the timeline into multiple segments, and each segment only contains a small amount of non-zero data. Algorithm 10 we present here is a slight modification of the algorithm in [39] by choosing a different partition threshold T_0, while the value of the partition threshold T_0 in [39] is originally related to an extra parameter, the confidence probability.

We clarify some terms used in the description of Algorithm 10:

- *Segment*: a segment is a time interval. Algorithm 10 partition the time interval $[0, T]$ into several sub-intervals called segments.
- *Online counting subroutine*: This subroutine takes a non-negative integer data stream and outputs a private running sum at each time t, as discussed in Section 5.2.1.

Algorithm 10 Sparse counting with a finite time horizon

Input: Time horizon $T < \infty$, Boolean data sequence $(X_t)_{t=1}^T$. Privacy parameter ε.
(Initialization) Set $T_0 = 9 \log T/\varepsilon$ to be the partition threshold and $j = 1$ denoting the number of segments. $t = 0$. Let $\widetilde{N} = 0$ denote the private count in the previous $(j-1)$ segments.
(Online counting subroutine) Start an online counting subroutine $\mathcal{A}_{\text{sub}}$ with input privacy parameter $\varepsilon/2$ and input data stream to be determined later.

while $t \leq T$ **do**
Set the segment count $N_j = 0$ and the private threshold $\widetilde{T}_j = T_0 + \lambda_j$, where $\lambda_j \sim \text{Lap}_{\mathbb{Z}}(2/\varepsilon)$.

while $t \leq T$ and $N_j + \lambda_t' \leq \widetilde{T}_j$, where $\lambda_t' \sim \text{Lap}_{\mathbb{Z}}(2/\varepsilon)$ **do**
Set $t \leftarrow t + 1$, $N_j \leftarrow N_j + X_t$.
Output. Output the same $\widetilde{N}$ for all t in this segment.
end while
End the segment j and set $j \leftarrow j + 1$.
Run $\mathcal{A}_{\text{sub}}$ with an updated input Boolean data stream $\{N_i\}_{i=1}^{j-1}$. Namely, N_{j-1} is added to the data stream. Update $\widetilde{N}$ to be the latest output of $\mathcal{A}_{\text{sub}}$.
end while

Algorithm 10 has various choices of the online counting subroutine $\mathcal{A}_{\text{sub}}$. One choice is the *Binary Mechanism* proposed in [22, 37], which ensures differential privacy for any input data

stream. The original version of [22, Algorithm 2] requires the data sequence to be Boolean, but it can also be used for non-negative integer data streams; see, for example, [39].

With the guarantee of the subroutine using the Binary Mechanism as in Lemma 5.4, [39] shows that their sparse counting Algorithm indeed improves the counting error. We will prove a similar expectation bound with the new partition threshold T_0 in Algorithm 10.

Lemma 5.6. *When choosing the online counting subroutine as the Binary Mechanism, Algorithm 10 attains ε-differential privacy. Let* error_t *denote the counting error between true count $\sum_{i=1}^{t} X_i$ and the output at time t. Then, for any fixed t, there is an accuracy bound*

$$\mathbb{E}\,\mathrm{error}_t \lesssim (\log T + \log^{1.5} n)/\varepsilon,$$

where T is the bound for the time horizon, and n is the number of the non-zero elements in the input data stream.

Proof. The privacy part follows from the original proof in [39, Theorem 3.1]. We focus on the accuracy guarantee.

The algorithm gives a private partition of the time interval and then treats each segment in the partition as a timestamp in the online counting subroutine. We will first prove that there are at most $n+1$ many segments in the partition.

Note that there are $2T$ many independent $\mathrm{Lap}(2/\varepsilon)$ random variables in total. Therefore, by a simple union bound argument, with probability $1/T$, their magnitudes are uniformly bounded by $B = \frac{2}{\varepsilon}(2\log T + \log 2)$. Conditioning on this event, we know $S_j + \mathrm{Lap}_{\mathbb{Z}}(2/\varepsilon) > \widetilde{T}_j$ implies that $\left|S_j - T_0\right| \le 2B$ and $S_j > T_0 - 2B > 0$. So whenever a segment is sealed, its true count is non-zero; hence, we have at most $n+1$ segments (in case the last one has not been sealed).

Now, we can compute the expectation of error. For the case where $2T$ many $\mathrm{Lap}(2/\varepsilon)$ random variables share the uniform bound B, the counting error consists of two parts: (1) the error from the online counting subroutine $\mathcal{A}_{\mathrm{sub}}$ and (2) the approximation error when ignoring the counts within time $[t_j, t]$ (i.e. S_j in the algorithm). The first part is bounded in (5.2), and the second error is bounded by $T_0 + 2B$ (as $S_j + \mathrm{Lap}_{\mathbb{Z}}(2/\varepsilon) \le \widetilde{T}_j$). So the total error is $O(\frac{1}{\varepsilon}(\log T + \log^{1.5}(n+1)))$. More precisely, the discussion can be written as the following inequality, where c_t and c_{t_j} denote corresponding true counts at time t, t_j:

$$\begin{aligned}
\left|c_t - \widetilde{S}\right| &\le \left|c_{t_j} - \widetilde{S}\right| + \left|c_t - c_{t_j}\right| \\
&= \left|c_{t_j} - \widetilde{S}\right| + S_j \\
&\le \left|c_{t_j} - \widetilde{S}\right| + T_0 + \left|S_j - T_0\right| \\
&\lesssim \frac{\log^{1.5}(n+1)}{\varepsilon} + \frac{\log T}{\varepsilon}.
\end{aligned}$$

For the other case, if the uniform upper bound fails with probability $1/T$, as the content of each segment is no longer available, we only have a trivial upper bound T for the number of segments. So the first error term from $\mathcal{A}_{\mathrm{sub}}$ becomes $O(\frac{1}{\varepsilon}\log^{1.5} T)$. And for the second part, we have a trivial upper bound $|S_j| \le n \le T$. Therefore, we have error $O(\frac{1}{\varepsilon}\log^{1.5} T + n)$.

By the law of total expectation, we have

$$\begin{aligned}
\mathbb{E}\,\mathrm{error}_t &\lesssim \Big(\frac{\log^{1.5}(n+1)}{\varepsilon} + \frac{\log T}{\varepsilon}\Big) + \frac{1}{T}\Big(\frac{1}{\varepsilon}\log^{1.5} T + n\Big) \\
&= O\Big(\frac{\log^{1.5}(n+1)}{\varepsilon} + \frac{\log T}{\varepsilon}\Big).
\end{aligned}$$

□

5.2.3 Inhomogeneous sparse counting

As a spoil to the online synthetic data algorithm, we will again consider the binary hierarchical partition of the hypercube as in Definition 2.7. In each subregion, our goal is to output a private count of the points $X_1, \ldots, X_t$ in Ω_θ, denoted by $n_\theta^{(t)}$. Since

$$n_\theta^{(t)} = \sum_{i=1}^{t} \mathbf{1}_{\{X_i \in \Omega_\theta\}},$$

this step is closely related to the differentially private online counting algorithms in this section. Furthermore, note that if Ω_θ is small and does not contain many X_i, then the Boolean series $\{\mathbf{1}_{\{X_i \in \Omega_\theta\}}\}$ also has sparse structure and Algorithm 10 could be applied.

Here, according to Section 5.1, note that the subregions are in different level, and smaller subregions might not exist at the beginning (see the main algorithm Algorithm 12 for more details), the subregions are asymmetric. Therefore, we introduce an inhomogeneity to the counting algorithm so we can optimize the noises scale later.

Algorithm 11 is based on Algorithm 10 and uses integer Laplacian noise with different variances in different time intervals. We now give several definitions to describe Algorithm 11:

- *Time level*: Starting from level 0, we say time t is at *level* j if $2^r/\varepsilon \leq t < 2^{r+1}/\varepsilon$. In Algorithm 11, we process the data stream level by level, where level r starts from the timestamp $t_r = \lceil 2^r/\varepsilon \rceil$.

- *Starting level*: We set an additional input r_0 to indicate the level from which the output starts. More precisely, the output of Algorithm 11 start from time $t_{r_0} = \lceil 2^{r_0}/\varepsilon \rceil$. We use $\widetilde{S}$ to store the private count from starting level r_0 to level $(r-1)$, and $\widetilde{S}$ does not include the count of data points arriving before the starting level r_0. This setting comes from the asymmetry of the subregions.

- *Counting subroutine*: The subroutine Algorithm 10 is an online counting algorithm with finite time horizon. It takes a Boolean data series as input and outputs the private counts of the first t data points at any time t. Here, we apply Algorithm 10 to obtain noisy counts c_t of the number of 1's arriving in the time interval $[t_r, t]$ for each $t \in [t_r, t_{r+1})$.

- *Update of $\widetilde{S}$*: During the counting subroutine, $\widetilde{S}$ is not updated. It is only updated at the end of the time level r by adding a noisy count $\sum_{t=t_r}^{t_{r+1}-1} X_t + \mathrm{Lap}_{\mathbb{Z}}(2/\varepsilon_r)$.

Algorithm 11 Inhomogeneous sparse counting

Input: Output starting level r_0. Boolean data sequence $\{X_t\}_{t=2^{r_0}}^{\infty}$. Noise parameters $\varepsilon_{r_0}, \varepsilon_{r_0+1}, \ldots$.
(Initialization) Set the finite private count $\widetilde{S} \leftarrow 0$, the current level $r \leftarrow r_0$, and the timestamps $t_{r_0} = \lceil 2^{r_0}/\varepsilon \rceil$, $t_{r_0+1} = \lceil 2^{r_0+1}/\varepsilon \rceil$.

while $r_0 \le r < \infty$ **do**
(Counting subroutine) For $t \in [t_r, t_{r+1})$, apply Algorithm 10 with time horizon $t_{r+1} - t_r$ and privacy parameter $\varepsilon_r/2$. Record the outputs $c_{t_r}, \ldots, c_{t_{r+1}-1}$.
Output. Output $\widetilde{S} + c_t$ as the private count at time t for each $t \in [t_r, t_{r+1})$.
Update

$$\widetilde{S} \leftarrow \widetilde{S} + \sum_{t=t_r}^{t_{r+1}-1} X_t + \mathrm{Lap}_{\mathbb{Z}}(2/\varepsilon_r)$$

and start a new level with $r \leftarrow r+1$, $t_{r+1} \leftarrow \lceil 2^{r+1}/\varepsilon \rceil$.
end while

The following lemma is a privacy guarantee for Algorithm 11. We show Algorithm 11 gives differential privacy under different notions of neighboring data sets $\mathcal{X}, \mathcal{X}'$ depending on when their different data points arrive in the data stream.

Lemma 5.7. *Let $\mathcal{A}$ be Algorithm 11. For two datasets $\mathcal{X}, \mathcal{X}'$ which differ on one data point at time $t \in [t_r, t_{r+1})$, and for any measurable subset S in the range of $\mathcal{A}$, the following holds:*

1. *If $r \ge r_0$, $\mathbb{P}\left\{\mathcal{A}(\mathcal{X}) \in S\right\} \le e^{\varepsilon_r} \cdot \mathbb{P}\left\{\mathcal{A}(\mathcal{X}') \in S\right\}$.*

2. *If* $r < r_0$, $\mathbb{P}\left\{\mathcal{A}(\mathcal{X}) \in S\right\} = \mathbb{P}\left\{\mathcal{A}(\mathcal{X}') \in S\right\}$.

Proof. For such $\mathcal{X}, \mathcal{X}'$ in the theorem, when $r < r_0$, one can notice that X_t does not appear in the algorithm. Therefore, the second assertion holds.

When $r \geq r_0$, to have the different data value at time t would make the following two influences:

1. When $\widetilde{S}$ first counts X_t privately applying the $\varepsilon_r/2$-differentially private subroutine;
2. When updating the count $\widetilde{S}$, we add noise $\mathrm{Lap}_{\mathbb{Z}}(2/\varepsilon_r)$, which implies another privacy budget $2/\varepsilon_r$.

By the parallel composition property of differential privacy [34], we know for the data at time t, the algorithm is ε_r differentially private. □

Lemma 5.8 bounds the difference between a noisy count and the true count in Algorithm 11 for different time intervals.

Lemma 5.8. *For each time* $t \in [t_r, t_{r+1})$, $\widetilde{S} + c_t$ *is the output of the noisy count at time* t *in Algorithm 11. We have*

$$\mathbb{E}\,|\widetilde{S} + c_t - S_t| \lesssim \sum_{i=1}^{t_{r_0}-1} X_i + \sum_{i=r_0}^{r-1} \frac{1}{\varepsilon_i} + \frac{\log t + \log^{1.5} n_r}{\varepsilon_r},$$

where $S_t := \sum_{i=1}^{t} X_i$ *is the true count at time* t *and* $n_r := \sum_{i=t_r}^{t_{r+1}} X_i$.

Proof. The accuracy part follows from the results of the subroutine and Laplacian mechanism. At time $t \in [t_r, t_{r+1})$, by the result of Lemma 5.6, we know the error of count c_t at

time t has bound

$$\mathbb{E}\left|c_t - \sum_{i=t_r}^{t} X_i\right| \lesssim \frac{\log t + \log^{1.5} n_r}{\varepsilon_r}.$$

And by the Laplacian mechanism, we know the count $\widetilde{S}$ has the accumulating error from each level (starting from r_0), namely

$$\mathbb{E}\left|\widetilde{S} - \sum_{i=t_{r_0}}^{t_r-1} X_i\right| \lesssim \sum_{i=r_0}^{r-1} \frac{1}{\varepsilon_i}.$$

Therefore, considering that $\widetilde{S}$ ignored the the data before level r_0, we deduce that

$$\begin{aligned}
\mathbb{E}|N_t - S_t| &= \mathbb{E}\left|\widetilde{S} + c_t - \sum_{i=1}^{t} X_i\right| \\
&\leq \sum_{i=1}^{t_{r_0}-1} X_i + \mathbb{E}\left|\widetilde{S} - \sum_{i=t_{r_0}}^{t_r-1} X_i\right| + \mathbb{E}\left|c_t - \sum_{i=t_r}^{t} X_i\right| \\
&\lesssim \sum_{i=1}^{t_{r_0}-1} X_i + \sum_{i=r_0}^{r-1} \frac{1}{\varepsilon_i} + \frac{\log t + \log^{1.5} n_r}{\varepsilon_r}.
\end{aligned}$$

□

5.3 Online synthetic data

5.3.1 Main algorithm

We can now introduce our main algorithm for online differentially private synthetic data release for the case, described formally in Algorithm 12.

Algorithm 12 uses the dynamic partition of the domain to generate synthetic data dynamically by adding dependent noise to perturb the counts of true data in each subregion. More

precisely, it consists of the following steps:

1. Refine a binary partition of $\Omega = [0,1]^d$ as time t grows. Equivalently, the tree $\mathcal{T}$ encoding the binary partition grows over time in the breath-first search order. We will refine the partition and create all sub-regions Ω_θ for all $|\theta| = j$ at timestamp $t_j = \lceil 2^j/\varepsilon \rceil$. We say t is of level j if $t_j \le t < t_{j+1}$, or equivalently $\mathcal{T}$ has depth j.

   Note that any sub-region Ω_θ with $|\theta| = j$ in Algorithm 12 only exists from level j: before level $|\theta|$ it was not created and after time t_{j+1} it still exists and will be refined.

2. For each existing region Ω_θ at time t, to count the number of data in Ω_θ privately, we output a perturbed count $N_\theta^{(t)}$ using the *Inhomogeneous Sparse Counting Algorithm* described in Algorithm 11. For each subregion Ω_θ, an online counting subroutine $\mathcal{A}_\theta$ starts as soon as Ω_θ is created, and it outputs a noisy count for every time t afterward. Privacy and accuracy guarantees for Algorithm 11 are given in Section 5.2.3.

   The choice of $r_0 = |\theta|$ for $d \ge 2$ indicates that Ω_θ will not load the historical data which came before level j (see Algorithm 11). The exact choices of the privacy parameters $\{\varepsilon_{j,r}\}$ in Algorithm 12 to implement the subroutines $\mathcal{A}_\theta$ when $d \ge 2$ are given in (5.3). The choices of $\varepsilon_{j,r}$ when $d = 1$ are given in (5.12).

3. The noisy counts in the Perturbation step could be negative and inconsistent. We post-process them to ensure they are non-negative, and the counts of subregions always add up to the count of the region in the upper level. The details of this step are given in Algorithm 3 from Chapter 3.

4. Finally, we turn the online synthetic counts in each region into online synthetic data by choosing the same amount of data points in each region whose location is independent of the true data.

Algorithm 12 Online synthetic data

Input: Privacy budget ε and an infinite data sequence $\{X_i\}_{i=1}^{\infty}$ where $X_i \in [0,1]^d$. For each time t, data points $(X_1, \ldots, X_t)$ are available.
(Initialization) Set $t = t_0 = 1$, $\Omega_\emptyset = \Omega$, and the depth of partition tree $r \leftarrow 0$.

while $t \in \mathbb{N}$ **do**
  while $t \geq 2^r/\varepsilon$ **do**
    $r \leftarrow r + 1$.
    (New binary partition) Partition Ω_θ into $\Omega_{\theta 0}$ and $\Omega_{\theta 1}$ for every $|\theta| = r - 1$.
    (New subroutines) For every newly created Ω_θ, Initiate a subroutine denoted by $\mathcal{A}_\theta$. Here $\mathcal{A}_\theta$ implements Algorithm 11 with input parameters given by starting level

$$r_0 = \begin{cases} |\theta| & \text{if } d \geq 2, \\ 0 & \text{if } d = 1 \end{cases}$$

and privacy parameters $\varepsilon_{|\theta|,r_0}, \varepsilon_{|\theta|,r_0+1}, \ldots$. The Input Boolean data sequence of $\mathcal{A}_\theta$ will be specified in the Perturbation step below.
  end while
  (Perturbation) For every Ω_θ, where $1 \leq |\theta| \leq r$, compute the noisy online count $N_\theta^{(t)}$ using the subroutine $\mathcal{A}_\theta$ with an updated input Boolean data stream $\left\{\mathbf{1}_{\{X_i \in \Omega_\theta\}}\right\}_{i=2^{|\theta|}}^{t}$. Namely, $\mathbf{1}_{\{X_t \in \Omega_\theta\}}$ is added to the data stream.
  (Consistency) Transform the perturbed counts of each subregion, $\{N_\theta^{(t)}\}_{|\theta| \leq r}$, into non-negative consistent counts $\{\widehat{N}_\theta^{(t)}\}_{|\theta| \leq r}$ using Algorithm 3.
  (Output) Output the synthetic data $\mathcal{Y}_t$ by choosing the locations of $\widehat{N}_\theta^{(t)}$ many data points arbitrarily and independently of the true data within each subregion Ω_θ where $|\theta| = r$.
  Let $t \leftarrow t + 1$.
end while

5.4 Proof of Theorem 5.1 when $d \geq 2$

We now prove that when $d \geq 2$, Algorithm 12 satisfies the privacy and accuracy guarantee in Theorem 5.1. To complete our proof, in Algorithms 12 and Algorithm 11, we choose privacy parameters

$$\varepsilon_{j,r} = C_1 \varepsilon 2^{(j-r)(1-1/d)/2}, \text{ where } C_1 = \frac{1 - 2^{-(1-1/d)/2}}{2}. \tag{5.3}$$

Denote $\alpha := 2^{(1-1/d)/2} \in [2^{1/4}, \sqrt{2})$ and we can check

$$\sum_{j=1}^{s} \varepsilon_{j,s} = \sum_{j=1}^{s} C_1 \varepsilon \alpha^{j-s} = \frac{C_1 \varepsilon (1 - \alpha^{-s})}{1 - \alpha^{-1}} \leq \frac{C_1 \varepsilon}{1 - \alpha^{-1}} \leq \frac{\varepsilon}{2}. \tag{5.4}$$

5.4.1 Privacy

Proposition 5.9. *For any choice of privacy parameters $\varepsilon_{j,r}$ satisfying*

$$\sup_{s \geq 0} \sum_{j=1}^{s} \varepsilon_{j,s} \leq \frac{\varepsilon}{2},$$

we have Algorithm 12 is ε-differentially private. In particular, with the choice of parameters in Equation (5.3)*, Algorithm 12 is ε-differentially private.*

Proof. Since the privacy budget in Algorithm 12 is only spent on the Perturbation step, we only need to show this step is ε-differentially private.

Consider two neighboring data sets $\mathcal{X}, \mathcal{X}'$, which are the same except for $X_t \in \mathcal{X}, X_t' \in \mathcal{X}'$ arriving at time t. Suppose the partition $\mathcal{T}$ at time t has depth $r = \lfloor \log_2(\varepsilon t) \rfloor$. Then, the true count of Ω_θ corresponding to $\mathcal{X}$ and $\mathcal{X}'$ are the same except for at most two subregions at each level in $\mathcal{T}$, and they form two paths of length r in the tree. For X_t, let us denote

these subregions

$$X_t \in \Omega_{\theta_r} \subset \cdots \subset \Omega_{\theta_1} \subset \Omega.$$

On the other hand, once X_t is given, we know exactly the corresponding subregions in $\mathcal{T}$ at level $r+1, r+2, \ldots$ will contain X_t in the future as soon as they are created. This gives us an infinite sequence of subregions

$$\Omega_{\theta_r} \supset \Omega_{\theta_{r+1}} \supset \cdots .$$

Similarly we can also obtain an infinite sequence $\Omega \supset \Omega_{\theta'_1} \supset \Omega_{\theta'_2} \supset \cdots$ containing X'_t.

Consider the first sequence. As the difference of $\mathcal{X}, \mathcal{X}'$ at time t will only influence the counts in Ω_{θ_j}, $j \geq 0$. We consider the subroutine $\mathcal{A}_{\theta_j}$ in each of the regions Ω_{θ_j} for all $j \geq 0$. There are two cases:

1. When $0 < j \leq r$, Ω_{θ_j} counts the data X_t at time t. So by Lemma 5.7, we protect the privacy of X_t with parameter $\varepsilon_{j,r}$.

2. When $j > r$, by the Initialization step in Algorithm 11, $\mathcal{A}_{\theta_j}$ and the private counts $N^{(t)}_{\theta_j}$ no longer depend on the value of X_t.

By Lemma 2.3, the parallel composition rule of differential privacy, and taking a supremum over all possible t, we have Algorithm 12 is differentially private with parameter

$$\sup_{s \geq 0} \sum_{j=1}^{s} \varepsilon_{j,s} \leq \frac{\varepsilon}{2},$$

where the inequality above holds due to (5.4).

The same argument holds for the second subregion sequence containing X'_t. Hence the whole

algorithm is ε-differentially private by applying the parallel composition rule, Lemma 2.3, again. $\square$

5.4.2 Accuracy

Now we consider the accuracy of the output in Wasserstein distance at time t with the corresponding level $r = \lfloor \log_2(\varepsilon t) \rfloor$. To ensure the accuracy of the consistency step, we include the following lemma from the proof to Theorem 3.9.

Lemma 5.10. *With data set $\mathcal{X}$ of size n and the binary partition structure $(\Omega_\theta)_{|\theta|\le r}$ of depth r, let λ_θ denote the noise adding to the true count in Ω_θ. Then by forcing consistency and generating synthetic data $\mathcal{Y}$ of size n according to the consistent private counts, the Wasserstein error is*

$$\mathbb{E}\, W_1(\mu_{\mathcal{X}}, \mu_{\mathcal{Y}}) \lesssim \frac{1}{n} \sum_{j=0}^{r-1} \sum_{|\theta|=j} \mathbb{E}\left[\max\{|\lambda_{\theta 0}|, |\lambda_{\theta 1}|\} \right] \operatorname{diam}(\Omega_\theta) + \delta. \tag{5.5}$$

Here $\delta = \max_{|\theta|=r} \operatorname{diam}(\Omega_\theta)$ is the maximal diameter of the subregions of depth r.

The proof to the lemma is referred to the proof to Theorem 3.9, where we first prove (5.5) and later substitute the noisy parameter of each noise λ_θ to get the accuracy bound for the general PMM algorithm.

We also need the following estimates in the proof.

Lemma 5.11. *Suppose $\alpha > 1$ is a constant and $r = \lfloor \log_2(\varepsilon t) \rfloor$, then*

$$S' = \sum_{j=1}^{r} \alpha^j (r - j + 1) \lesssim \alpha^r,$$

$$S = \sum_{j=1}^{r} \alpha^j (r - j + 1)^2 \lesssim \alpha^r.$$

Proof. We have the following holds:

$$\begin{aligned}
\alpha S' &= \sum_{j=1}^{r} \alpha^{j+1}(r-j+1) = \sum_{j=2}^{r+1} \alpha^{j}(r-j+2), \\
(\alpha-1)S' &= \alpha S - S = \sum_{j=2}^{r+1} \alpha^{j} + \alpha^{r+1} - \alpha r, \\
\implies S' &= \frac{\alpha^{r+1}-\alpha^2}{\alpha-1} - \alpha r \lesssim \alpha^r. \qquad (5.6)
\end{aligned}$$

For the sum S, again, there is

$$\begin{aligned}
\alpha S &= \sum_{j=1}^{r} \alpha^{j+1}(r-j+1)^2 = \sum_{j=2}^{r+1} \alpha^{j}(r-j+2)^2, \\
(\alpha-1)S &= \alpha S - S = \sum_{j=2}^{r+1} \alpha^{j}(2r-2j+3) + \alpha^{r+1} - \alpha r^2.
\end{aligned}$$

Applying (5.6), we have

$$S = \frac{1}{\alpha-1}\left(\sum_{j=2}^{r+1} \alpha^{j}(2r-2j+3) + \alpha^{r+1} + \alpha r^2\right) \lesssim \alpha^r + \alpha^{r+1} - \alpha r^2 \lesssim \alpha^r$$

as desired. □

Next we are ready to prove the accuracy bound (5.1) for $d \geq 2$.

Proof of (5.1) *for* $d \geq 2$. Let $\lambda_\theta^{(t)} = N_\theta^{(t)} - n_\theta^{(t)}$ be the counting noise of subregion Ω_θ at time t with $1 \leq j = |\theta| \leq r$, where $t_r \leq t < t_{r+1}$. Recall that t_i, defined as $t_i = \lceil 2^i/\varepsilon \rceil$, denotes the timestamp when level i starts. Note that Ω_θ is created at time level j, which is also the starting level parameter in subroutine $\mathcal{A}_\theta$.

By Lemma 5.8, we have the upper bound of the noise at time $t \geq \frac{e}{\varepsilon}$ for Ω_θ,

$$\mathbb{E}\left|\lambda_\theta^{(t)}\right| \lesssim \left|\{X_s \mid s < t_j, X_s \in \Omega_\theta\}\right| + \sum_{i=j}^{r-1} \frac{1}{\varepsilon_{j,i}} + \frac{\log t + \log^{1.5} n_j}{\varepsilon_{j,r}}. \tag{5.7}$$

Applying Lemma 5.10, at a fixed time $t \geq \frac{e}{\varepsilon}$, we have

$$\mathbb{E}\, W_1(\mu_{\mathcal{X}_t}, \mu_{\mathcal{Y}_t}) \leq \frac{1}{t}\left[\sum_{j=0}^{r-1} \sum_{\theta\in\{0,1\}^j} \mathbb{E}\left[\max\left\{\left|\lambda_{\theta 0}^{(t)}\right|, \left|\lambda_{\theta 1}^{(t)}\right|\right\}\right] \operatorname{diam}(\Omega_\theta)\right] + \delta, \tag{5.8}$$

where $\delta = \max_{|\theta|=r} \operatorname{diam}(\Omega_\theta)$ denotes the maximal diameter of the subregions of depth r. For $\Omega = [0,1]^d$, we have $\operatorname{diam}(\Omega_\theta) \asymp 2^{-|\theta|/d}$ and there are $2^{|\theta|}$ many different subregions of such size.

Note that for fixed j, $\varepsilon_{j,i}$ decreases as i increases. From (5.8) and (5.7), we have for $t \geq \frac{e}{\varepsilon}$,

$$\begin{aligned}
\mathbb{E}\, W_1(\mu_{\mathcal{X}_t}, \mu_{\mathcal{Y}_t}) &\leq \frac{1}{t}\sum_{j=0}^{r} \sum_{|\theta|=j} \mathbb{E}\left|\lambda_\theta^{(t)}\right| \cdot 2^{-j/d} + 2^{-r/d} \\
&\lesssim \frac{1}{t}\sum_{j=0}^{r} \sum_{|\theta|=j} \left(\left|\{X_s \mid s < t_j, X_s \in \Omega_\theta\}\right| + \right. \\
&\qquad \left. \sum_{i=j}^{r-1} \frac{1}{\varepsilon_{j,i}} + \frac{\log t + \log^{1.5}(n_\theta^{(t)} + 1)}{\varepsilon_{j,r}}\right) \cdot 2^{-j/d} + 2^{-r/d} \\
&\lesssim \frac{1}{t}\sum_{j=0}^{r} \left(\frac{2^j}{\varepsilon} + \sum_{|\theta|=j} \frac{\log t + \log^{1.5}(n_\theta^{(t)} + 1)}{\varepsilon_{j,r}}\right) \cdot 2^{-j/d} + 2^{-r/d}
\end{aligned} \tag{5.9}$$

Since

$$\left(\log^{1.5} x\right)'' = \frac{1.5\left(\frac{1}{2\sqrt{\log x}} - \sqrt{\log x}\right)}{x^2} \leq 0, \quad \text{if } x \geq \sqrt{e},$$

the function $\log^{1.5}(x+2)$ is concave on $[0, +\infty)$. Therefore, for fixed j, we can apply Jensen's

inequality when summing the $\log^{1.5}(n_\theta^{(t)}+1)$ terms over all $|\theta|=j$ and obtain

$$\begin{aligned}\sum_{|\theta|=j}\log^{1.5}(n_\theta^{(t)}+1) &\le \sum_{|\theta|=j}\log^{1.5}(n_\theta^{(t)}+2)\\ &\le 2^j\log^{1.5}\Big(2^{-j}\sum_{|\theta|=j}n_\theta^{(t)}+2\Big)\\ &\le 2^j\log^{1.5}\Big(\frac{t}{2^j}+2\Big).\end{aligned}$$

Substitute the result above into (5.9) and we have

$$\mathbb{E}\,W_1(\mu_{\mathcal{X}_t},\mu_{\mathcal{Y}_t})\lesssim\frac{2^{r(1-1/d)}}{\varepsilon t}+2^{-r/d}+\frac{1}{t}\sum_{j=0}^{r}\frac{1}{\varepsilon_{j,r}}\Big(\log t+\log^{1.5}\Big(\frac{t}{2^j}+2\Big)\Big)2^{j(1-1/d)}\quad(5.10)$$

As $t\in[t_r,t_{r+1})$, we have

$$\begin{aligned}\log(t/2^j+2)&\le\log(2^{r-j+1}/\varepsilon+2)\\ &=\log(2^{r-j+1})+\log\Big(\frac{1}{\varepsilon}+2^{j-r}\Big)\\ &\lesssim(r-j+1)+\log\Big(\frac{1}{\varepsilon}+1\Big).\end{aligned}$$

To attain ε-differentially privacy, we can choose the privacy parameters to optimize the accuracy in Wasserstein distance given in (5.10). One of the nearly best choices is given in (5.3). Therefore, for the second term in (5.10), we deduce that

$$\begin{aligned}&\sum_{j=1}^{r}\frac{1}{\varepsilon_{j,r}}\Big(\log t+\log^{1.5}\Big(\frac{t}{2^j}+2\Big)\Big)2^{j(1-1/d)}\\ \lesssim&\frac{2^{r(1-1/d)/2}}{\varepsilon}\cdot\sum_{j=1}^{r}\Big(\log t+(r-j+1)^2+\log^2\Big(\frac{1}{\varepsilon}+1\Big)\Big)2^{j(1-1/d)/2}\\ \lesssim&\frac{2^{r(1-1/d)}}{\varepsilon}\Big(\log t+\log^2\Big(\frac{1}{\varepsilon}+1\Big)\Big).\end{aligned}\quad(5.11)$$

Here, the last inequality uses Lemma 5.11 with $\alpha = 2^{\frac{1-1/d}{2}} > 1$ when $d \geq 2$.

Therefore, when $d \geq 2$, with (5.10) and (5.11) we have

$$\begin{aligned}\mathbb{E}\, W_1(\mu_{\mathcal{X}_t}, \mu_{\mathcal{Y}_t}) &\lesssim \frac{2^{r(1-1/d)}}{\varepsilon t} + \frac{\log t + \log^2\left(\frac{1}{\varepsilon}+1\right)}{\varepsilon t} \cdot 2^{r(1-1/d)} + (\varepsilon t)^{-1/d} \\ &\lesssim \left(\log t + \log^2\left(\frac{1}{\varepsilon}+1\right)\right) \cdot (\varepsilon t)^{-1/d}.\end{aligned}$$

When $t \geq e/\varepsilon + \exp(\log^2(1/\varepsilon + 1))$, the inequality above can be simplified as follows

$$\mathbb{E}\, W_1(\mu_{\mathcal{X}_t}, \mu_{\mathcal{Y}_t}) \lesssim \log(t)(\varepsilon t)^{-1/d}.$$

This finishes the proof. □

5.5 Proof of Theorem 5.1 for $d = 1$

We now prove the privacy and accuracy guarantee of Algorithm 12 for $d = 1$ with a different choice of privacy parameters $\varepsilon_{j,r}$ given in (5.12).

Proposition 5.12. *When $d = 1$, Algorithm 12 satisfies ε-online privacy with privacy parameters*

$$\varepsilon_{j,r} = \frac{3}{\pi^2} \cdot \frac{\varepsilon}{(j+1)^2} \tag{5.12}$$

for all $r \geq j$. Moreover, for any time $t \geq e/\varepsilon$,

$$\mathbb{E}W_1(\mu_{\mathcal{X}_t}, \mu_{\mathcal{Y}_t}) \lesssim \frac{1}{\varepsilon t} \log^3(\varepsilon t) \log^{1.5} t.$$

Proof. The privacy guarantee follows from the proof to Proposition 5.9. Since we have

$r_0 = 0$ when $d = 1$, for every data X_t, it influences the true count of Ω_θ for exactly one $\theta = \theta_j$ with $|\theta| = j, j \geq 1$. Therefore, following the the proof to Proposition 5.9, we know the Algorithm 12 with $d = 1$ is ε-differentially private as

$$2 \sup_{s \geq 1} \sum_{j=1}^{\infty} \varepsilon_{j,s} = \frac{6}{\pi^2} \sum_{j=1}^{\infty} \frac{\varepsilon}{(j+1)^2} = \varepsilon.$$

The accuracy in Wasserstein distance follows from the accuracy proof to Theorem 5.1 in Section 5.4. Note that when $d = 1$, we apply the counting subroutine Algorithm 11 with parameter $r_0 = 0$, which means we do not ignore the earlier data for any Ω_θ. Therefore, in (5.7), there is no longer the first term, which indicates the number of data we ignore for Ω_θ. Hence, following the proof to (5.10), at any time t, the Wasserstein error between the true data set $\mathcal{X}_t$ and the synthetic data set $\mathcal{Y}_t$ at time t is

$$\begin{aligned}
&\mathbb{E}\, W_1(\mu_{\mathcal{X}_t}, \mu_{\mathcal{Y}_t}) \\
&\lesssim \frac{1}{t} \sum_{j=0}^{r} \frac{1}{\varepsilon_{j,r}} \Big(\log t + \log^{1.5} \Big(\frac{t}{2^j} + 2 \Big) \Big) 2^{j(1-1/d)} + 2^{-r/d},
\end{aligned}$$

where r is the level at time t, i.e., $\lceil 2^r/\varepsilon \rceil \leq t < \lceil 2^{r+1}/\varepsilon \rceil$. Substitute $d = 1$ and the new choices of privacy parameters, we have

$$\begin{aligned}
&\mathbb{E}\, W_1(\mu_{\mathcal{X}_t}, \mu_{\mathcal{Y}_t}) \\
&\lesssim \frac{1}{t} \sum_{j=0}^{r} \frac{1}{\varepsilon_{j,r}} \Big(\log t + \log^{1.5} \Big(\frac{t}{2^j} + 2 \Big) \Big) + 2^{-r} \\
&\lesssim \frac{1}{t} \sum_{j=0}^{r} \frac{(j+1)^2}{\varepsilon} \Big(\log t + \log^{1.5} \Big(\frac{t}{2^j} + 2 \Big) \Big) + 2^{-r} \\
&\lesssim \frac{1}{t} \sum_{j=0}^{r} \frac{(j+1)^2}{\varepsilon} \log^{1.5} t + \frac{1}{\varepsilon t} \\
&\lesssim \frac{(r+1)^3 \log^{1.5} t}{\varepsilon t}
\end{aligned}$$

$$\lesssim \frac{(\log(\varepsilon t)+1)^3 \log^{1.5} t}{\varepsilon t}.$$

When $t \geq e/\varepsilon$, the inequality above becomes

$$\mathbb{E}\, W_1(\mu_{\mathcal{X}_t}, \mu_{\mathcal{Y}_t}) \lesssim \frac{\log^3(\varepsilon t)\log^{1.5}(t)}{\varepsilon t}.$$

This finishes the proof. □

5.6 Time complexity

We consider the running time for the algorithms to output after the input data arrives at a fixed timestamp t. The accumulating time complexity of the algorithms for time $1, \ldots, t$ need a further sum of the time complexity at a fixed timestamp.

The Binary Mechanism in [37, 22] has time complexity $O(\log t)$ to give the output at time t, as it sums over $O(\log t)$ many Laplacian random variables. Same time complexity holds for sparse counting Algorithm 10, as it checks the partition threshold with $O(1)$ time and runs Binary Mechanism as a subroutine.

As for Algorithm 11, *Inhomogeneous Sparse Counting*, it is connected by multiple implementations of Algorithm 10. Furthermore, for a given $t \in [t_r, t_{r+1})$, only one such subroutine is active with time horizon $t_{r+1} - t_r \asymp 2^r/\varepsilon \asymp t$, so the time complexity is also $O(\log t)$.

For our main Algorithm 12, there is another variable d, the data dimension. We can decompose the procedure of Algorithm 12 at time t as the following steps:

- (Partition) For each fixed time t in Algorithm 12, there are $O(\varepsilon t)$ many sub-regions Ω_θ in the binary partition tree of $[0,1]^d$, and when further partition happens, $O(\varepsilon t)$ many

subregions are created.

- (Perturbation) Whenever a new data comes at timestamp t, it takes $O(\log(\varepsilon t))$ complexity to determine the subregions where the new data belongs. Afterwards, for all subroutines $\mathcal{A}_\theta$'s, they in total take running time $O(\varepsilon t \log t)$ by the result above for Algorithm 11.

- (Consistency and Output) The time complexity is $O(\varepsilon t)$ for the consistency step and $O(dt)$ for the output, as the output is d-dimensional data set of size t.

Therefore, the whole Algorithm 12 has time complexity $O(dt + \varepsilon t \log t)$ to output at time t.

Bibliography

[1] J. Abowd, R. Ashmead, G. Simson, D. Kifer, P. Leclerc, A. Machanavajjhala, and W. Sexton. Census topdown: Differentially private data, incremental schemas, and consistency with public knowledge. *US Census Bureau*, 2019.

[2] J. M. Abowd, R. Ashmead, R. Cumings-Menon, S. Garfinkel, M. Heineck, C. Heiss, R. Johns, D. Kifer, P. Leclerc, A. Machanavajjhala, et al. The 2020 census disclosure avoidance system topdown algorithm. *Harvard Data Science Review*, Special Issue 2, 2022.

[3] D. Achlioptas and F. McSherry. Fast computation of low-rank matrix approximation. In *Proceedings of the thirty-P third annual ACM symposium on Theory of computing*, pages 611–618. ACM, 2001.

[4] K. Amin, T. Dick, A. Kulesza, A. Munoz, and S. Vassilvitskii. Differentially private covariance estimation. *Advances in Neural Information Processing Systems*, 32, 2019.

[5] R. Arora, J. Upadhyay, et al. Differentially private robust low-rank approximation. *Advances in neural information processing systems*, 31, 2018.

[6] K. D. Ba, H. L. Nguyen, H. N. Nguyen, and R. Rubinfeld. Sublinear time algorithms for earth movers distance. *Theory of Computing Systems*, 48:428–442, 2011.

[7] M. Balog, I. Tolstikhin, and B. Schölkopf. Differentially private database release via kernel mean embeddings. In *International Conference on Machine Learning*, pages 414–422. PMLR, 2018.

[8] B. Barak, K. Chaudhuri, C. Dwork, S. Kale, F. McSherry, and K. Talwar. Privacy, accuracy, and consistency too: a holistic solution to contingency table release. In *Proceedings of the twenty-sixth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 273–282, 2007.

[9] P. L. Bartlett and S. Mendelson. Rademacher and Gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3(Nov):463–482, 2002.

[10] R. Bhatia. *Matrix analysis*, volume 169. Springer Science & Business Media, 2013.

[11] A. Blum, K. Ligett, and A. Roth. A learning theory approach to noninteractive database privacy. *Journal of the ACM (JACM)*, 60(2):1–25, 2013.

[12] S. G. Bobkov and M. Ledoux. A simple fourier analytic proof of the AKT optimal matching theorem. *The Annals of Applied Probability*, 31(6):2567–2584, 2021.

[13] M. Boedihardjo, T. Strohmer, and R. Vershynin. Covariances loss is privacys gain: Computationally efficient, private and accurate synthetic data. *Foundations of Computational Mathematics*, pages 1–48, 2022.

[14] M. Boedihardjo, T. Strohmer, and R. Vershynin. Private measures, random walks, and synthetic data. *arXiv preprint arXiv:2204.09167*, 2022.

[15] M. Boedihardjo, T. Strohmer, and R. Vershynin. Private sampling: a noiseless approach for generating differentially private synthetic data. *SIAM Journal on Mathematics of Data Science*, 4(3):1082–1115, 2022.

[16] M. Boedihardjo, T. Strohmer, and R. Vershynin. Covariance loss, Szemeredi regularity, and differential privacy. *arXiv preprint arXiv:2301.02705*, 2023.

[17] J. Bolot, N. Fawaz, S. Muthukrishnan, A. Nikolov, and N. Taft. Private decayed predicate sums on streams. In *Proceedings of the 16th International Conference on Database Theory*, pages 284–295, 2013.

[18] S. Bubeck and M. Sellke. A universal law of robustness via isoperimetry. *Advances in Neural Information Processing Systems*, 34:28811–28822, 2021.

[19] M. Bun, T. Steinke, and J. Ullman. Make up your mind: The price of online queries in differential privacy. In *Proceedings of the twenty-eighth annual ACM-SIAM symposium on discrete algorithms*, pages 1306–1325. SIAM, 2017.

[20] M. Bun, M. Gaboardi, M. Neunhoeffer, and W. Zhang. Continual release of differentially private synthetic data from longitudinal data collections. *Proceedings of the ACM on Management of Data*, 2(2):1–26, 2024.

[21] A. R. Cardoso and R. Rogers. Differentially private histograms under continual observation: Streaming selection into the unknown. In *International Conference on Artificial Intelligence and Statistics*, pages 2397–2419. PMLR, 2022.

[22] T.-H. H. Chan, E. Shi, and D. Song. Private and continual release of statistics. *ACM Transactions on Information and System Security (TISSEC)*, 14(3):1–24, 2011.

[23] K. Chaudhuri, C. Monteleoni, and A. D. Sarwate. Differentially private empirical risk minimization. *Journal of Machine Learning Research*, 12(3), 2011.

[24] K. Chaudhuri, A. D. Sarwate, and K. Sinha. A near-optimal algorithm for differentially-private principal components. *Journal of Machine Learning Research*, 14, 2013.

[25] Y. Chen, A. Machanavajjhala, M. Hay, and G. Miklau. Pegasus: Data-adaptive differentially private stream processing. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 1375–1388, 2017.

[26] R. Cummings, S. Krehbiel, K. A. Lai, and U. Tantipongpipat. Differential privacy for growing databases. *Advances in Neural Information Processing Systems*, 31, 2018.

[27] G. Dai, Z. Su, and H. Wang. Tail bounds on the spectral norm of sub-exponential random matrices. *Random Matrices: Theory and Applications*, 13(01):2350013, 2024.

[28] C. Davis and W. M. Kahan. The rotation of eigenvectors by a perturbation. iii. *SIAM Journal on Numerical Analysis*, 7(1):1–46, 1970.

[29] S. Dereich, M. Scheutzow, and R. Schottstedt. Constructive quantization: Approximation by empirical measures. *Annales de l'IHP Probabilités et statistiques*, 49(4): 1183–1203, 2013.

[30] S. Dirksen. Tail bounds via generic chaining. *Electron. J. Probab*, 20(53):1–29, 2015.

[31] J. Domingo-Ferrer, D. Sánchez, and A. Blanco-Justicia. The limits of differential privacy (and its misuse in data release and machine learning). *Communications of the ACM*, 64(7):33–35, 2021.

[32] W. Dong, Y. Liang, and K. Yi. Differentially private covariance revisited. *Advances in Neural Information Processing Systems*, 35:850–861, 2022.

[33] K. Donhauser, J. Lokna, A. Sanyal, M. Boedihardjo, R. Hönig, and F. Yang. Certified private data release for sparse lipschitz functions. In *International Conference on Artificial Intelligence and Statistics*, pages 1396–1404. PMLR, 2024.

[34] C. Dwork and A. Roth. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014.

[35] C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov, and M. Naor. Our data, ourselves: Privacy via distributed noise generation. In *Advances in Cryptology-EUROCRYPT 2006: 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, St. Petersburg, Russia, May 28-June 1, 2006. Proceedings 25*, pages 486–503. Springer, 2006.

[36] C. Dwork, M. Naor, O. Reingold, G. N. Rothblum, and S. Vadhan. On the complexity of differentially private data release: efficient algorithms and hardness results. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*, pages 381–390, 2009.

[37] C. Dwork, M. Naor, T. Pitassi, and G. N. Rothblum. Differential privacy under continual observation. In *Proceedings of the forty-second ACM symposium on Theory of computing*, pages 715–724, 2010.

[38] C. Dwork, K. Talwar, A. Thakurta, and L. Zhang. Analyze Gauss: optimal bounds for privacy-preserving principal component analysis. In *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*, pages 11–20, 2014.

[39] C. Dwork, M. Naor, O. Reingold, and G. N. Rothblum. Pure differential privacy for rectangle queries via private partitions. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 735–751. Springer, 2015.

[40] C. Dwork, A. Nikolov, and K. Talwar. Efficient algorithms for privately releasing marginals via convex relaxations. *Discrete & Computational Geometry*, 53:650–673, 2015.

[41] C. Dwork, N. Kohli, and D. Mulligan. Differential privacy in practice: Expose your epsilons! *Journal of Privacy and Confidentiality*, 9(2), 2019.

[42] A. Epasto, J. Mao, A. M. Medina, V. Mirrokni, S. Vassilvitskii, and P. Zhong. Differentially private continual releases of streaming frequency moment estimations. In *14th Innovations in Theoretical Computer Science Conference (ITCS 2023)*. Schloss-Dagstuhl-Leibniz Zentrum für Informatik, 2023.

[43] U. Feige and E. Ofek. Spectral techniques applied to sparse random graphs. *Random Structures & Algorithms*, 27(2):251–275, 2005.

[44] H. Fichtenberger, M. Henzinger, and J. Upadhyay. Constant matters: Fine-grained error bound on differentially private continual observation. In *International Conference on Machine Learning*, pages 10072–10092. PMLR, 2023.

[45] D. J. Foster and A. Rakhlin. ℓ^∞ vector contraction for Rademacher complexity. *arXiv preprint arXiv:1911.06468*, 6, 2019.

[46] A. J. George, L. Ramesh, A. V. Singh, and H. Tyagi. Continual mean estimation under user-level privacy. *IEEE Journal on Selected Areas in Information Theory*, 2024.

[47] L.-A. Gottlieb, A. Kontorovich, and R. Krauthgamer. Adaptive metric dimensionality reduction. *Theoretical Computer Science*, 620:105–118, 2016.

[48] A. Guha Thakurta and A. Smith. (nearly) optimal algorithms for private online learning in full-information and bandit settings. *Advances in Neural Information Processing Systems*, 26, 2013.

[49] F. Harder, K. Adamczewski, and M. Park. Dp-merf: Differentially private mean embeddings with random features for practical privacy-preserving data generation. In *International conference on artificial intelligence and statistics*, pages 1819–1827. PMLR, 2021.

[50] M. Hardt and E. Price. The noisy power method: A meta algorithm with applications. *Advances in neural information processing systems*, 27, 2014.

[51] M. Hardt and A. Roth. Beyond worst-case analysis in private singular vector computation. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pages 331–340, 2013.

[52] M. Hardt, K. Ligett, and F. McSherry. A simple and practical algorithm for differentially private data release. *Advances in neural information processing systems*, 25, 2012.

[53] Y. He, R. Vershynin, and Y. Zhu. Algorithmically effective differentially private synthetic data. In G. Neu and L. Rosasco, editors, *Proceedings of Thirty Sixth Conference on Learning Theory*, volume 195 of *Proceedings of Machine Learning Research*, pages 3941–3968. PMLR, 12–15 Jul 2023.

[54] M. Henzinger, J. Upadhyay, and S. Upadhyay. Almost tight error bounds on differentially private continual counting. In *Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 5003–5039. SIAM, 2023.

[55] M. Henzinger, J. Upadhyay, and S. Upadhyay. A unifying framework for differentially private sums under continual observation. In *Proceedings of the 2024 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 995–1018. SIAM, 2024.

[56] H. Imtiaz and A. D. Sarwate. Symmetric matrix perturbation for differentially-private principal component analysis. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2339–2343. IEEE, 2016.

[57] S. Inusah and T. J. Kozubowski. A discrete analogue of the laplace distribution. *Journal of statistical planning and inference*, 136(3):1090–1102, 2006.

[58] P. Jain, P. Kothari, and A. Thakurta. Differentially private online learning. In *Conference on Learning Theory*, pages 24–1. JMLR Workshop and Conference Proceedings, 2012.

[59] P. Jain, C. Jin, S. M. Kakade, P. Netrapalli, and A. Sidford. Streaming PCA: Matching matrix Bernstein and near-optimal finite sample guarantees for Ojas algorithm. In *Conference on learning theory*, pages 1147–1164. PMLR, 2016.

[60] P. Jain, S. Raskhodnikova, S. Sivakumar, and A. Smith. The price of differential privacy under continual observation. In *International Conference on Machine Learning*, pages 14654–14678. PMLR, 2023.

[61] W. Jiang, C. Xie, and Z. Zhang. Wishart mechanism for differentially private principal components analysis. *Proceedings of the AAAI Conference on Artificial Intelligence*, 30(1), 2016.

[62] X. Jiang, Z. Ji, S. Wang, N. Mohammed, S. Cheng, and L. Ohno-Machado. Differential-private data publishing through component analysis. *Transactions on data privacy*, 6 (1):19, 2013.

[63] P. Kairouz, B. McMahan, S. Song, O. Thakkar, A. Thakurta, and Z. Xu. Practical and private (deep) learning without sampling or shuffling. In *International Conference on Machine Learning*, pages 5213–5225. PMLR, 2021.

[64] G. Kamath, J. Li, V. Singhal, and J. Ullman. Privately learning high-dimensional distributions. In *Conference on Learning Theory*, pages 1853–1902. PMLR, 2019.

[65] M. Kapralov and K. Talwar. On differentially private low rank approximation. In *Proceedings of the twenty-fourth annual ACM-SIAM symposium on Discrete algorithms*, pages 1395–1414. SIAM, 2013.

[66] J. W. Kim, K. Edemacu, J. S. Kim, Y. D. Chung, and B. Jang. A survey of differential privacy-based techniques and their applicability to location-based services. *Computers & Security*, 111:102464, 2021.

[67] L. V. Kovalev. Lipschitz clustering in metric spaces. *The Journal of Geometric Analysis*, 32(7):188, 2022.

[68] E. Kreacic, N. Nouri, V. K. Potluru, T. Balch, and M. Veloso. Differentially private synthetic data using kd-trees. In *The 39th Conference on Uncertainty in Artificial Intelligence*, 2023.

[69] G. Kumar, T. Strohmer, and R. Vershynin. An algorithm for streaming differentially private data. *arXiv preprint arXiv:2401.14577*, 2024.

[70] X. Li, S. Wang, and Y. Cai. Tutorial: Complexity analysis of singular value decomposition and its variants. *arXiv preprint arXiv:1906.12085*, 2019.

[71] T. Liu, G. Vietri, and S. Z. Wu. Iterative methods for private synthetic data: Unifying framework and new methods. *Advances in Neural Information Processing Systems*, 34: 690–702, 2021.

[72] X. Liu, W. Kong, S. Kakade, and S. Oh. Robust and differentially private mean estimation. *Advances in neural information processing systems*, 34:3887–3901, 2021.

[73] X. Liu, W. Kong, P. Jain, and S. Oh. DP-PCA: Statistically optimal and differentially private pca. *Advances in neural information processing systems*, 35:29929–29943, 2022.

[74] X. Liu, W. Kong, and S. Oh. Differential privacy and robust statistics in high dimensions. In *Conference on Learning Theory*, pages 1167–1246. PMLR, 2022.

[75] O. Mangoubi and N. Vishnoi. Re-analyze Gauss: Bounds for private matrix approximation via Dyson Brownian motion. *Advances in Neural Information Processing Systems*, 35:38585–38599, 2022.

[76] S. Mendelson. Empirical processes with a bounded ψ 1 diameter. *Geometric and Functional Analysis*, 20(4):988–1027, 2010.

[77] L. Meunier, B. J. Delattre, A. Araujo, and A. Allauzen. A dynamical system perspective for Lipschitz neural networks. In *International Conference on Machine Learning*, pages 15484–15500. PMLR, 2022.

[78] D. Mir, S. Muthukrishnan, A. Nikolov, and R. N. Wright. Pan-private algorithms via statistics on sketches. In *Proceedings of the thirtieth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 37–48, 2011.

[79] E. Oja. Simplified neuron model as a principal component analyzer. *Journal of mathematical biology*, 15:267–273, 1982.

[80] L. Otten. Latex template for thesis and dissertation documents at uc irvine. https://github.com/lotten/uci-thesis-latex/, 2012.

[81] Y. Qiu and K. Yi. Differential privacy on dynamic data. *arXiv preprint arXiv:2209.01387*, 2022.

[82] M. Reiss and M. Wahl. Nonasymptotic upper bounds for the reconstruction error of PCA. *The Annals of Statistics*, 48(2):1098–1123, 2020.

[83] V. Singhal and T. Steinke. Privately learning subspaces. *Advances in Neural Information Processing Systems*, 34:1312–1324, 2021.

[84] S. Song, S. Little, S. Mehta, S. Vinterbo, and K. Chaudhuri. Differentially private continual release of graph statistics. *arXiv preprint arXiv:1809.02575*, 2018.

[85] M. Talagrand. Matching random samples in many dimensions. *The Annals of Applied Probability*, pages 846–856, 1992.

[86] M. Talagrand. *The generic chaining: upper and lower bounds of stochastic processes*. Springer Science & Business Media, 2005.

[87] J. Thaler, J. Ullman, and S. Vadhan. Faster algorithms for privately releasing marginals. In *Automata, Languages, and Programming: 39th International Colloquium, ICALP 2012, Warwick, UK, July 9-13, 2012, Proceedings, Part I 39*, pages 810–821. Springer, 2012.

[88] V. Tikhomirov. ε-entropy and ε-capacity of sets in functional spaces. In *Selected works of AN Kolmogorov*, pages 86–170. Springer, 1993.

[89] J. Ullman and S. Vadhan. PCPs and the hardness of generating private synthetic data. In *Theory of Cryptography: 8th Theory of Cryptography Conference, TCC 2011, Providence, RI, USA, March 28-30, 2011. Proceedings 8*, pages 400–416. Springer, 2011.

[90] S. Vadhan. The complexity of differential privacy. *Tutorials on the Foundations of Cryptography: Dedicated to Oded Goldreich*, pages 347–450, 2017.

[91] V. V. Vazirani. *Approximation algorithms*, volume 1. Springer, 2001.

[92] R. Vershynin. *High-dimensional probability: An introduction with applications in data science*, volume 47. Cambridge university press, 2018.

[93] G. Vietri, C. Archambeau, S. Aydore, W. Brown, M. Kearns, A. Roth, A. Siva, S. Tang, and S. Wu. Private synthetic data for multitask learning and marginal queries. In *Advances in Neural Information Processing Systems*, 2022.

[94] C. Villani. *Optimal transport: old and new*, volume 338. Springer, 2009.

[95] U. von Luxburg and O. Bousquet. Distance-based classification with Lipschitz functions. *J. Mach. Learn. Res.*, 5(Jun):669–695, 2004.

[96] Z. Wang, C. Jin, K. Fan, J. Zhang, J. Huang, Y. Zhong, and L. Wang. Differentially private data releasing for smooth queries. *The Journal of Machine Learning Research*, 17(1):1779–1820, 2016.

[97] J. Weed and F. Bach. Sharp asymptotic and finite-sample rates of convergence of empirical measures in wasserstein distance. *Bernoulli*, 25(4 A):2620–2648, 2019.

[98] Y. Yang, K. Adamczewski, D. J. Sutherland, X. Li, and M. Park. Differentially private neural tangent kernels for privacy-preserving data generation. *arXiv preprint arXiv:2303.01687*, 2023.

[99] Y. Yu, T. Wang, and R. J. Samworth. A useful variant of the Davis–Kahan theorem for statisticians. *Biometrika*, 102(2):315–323, 2015.

[100] S. Zhou, K. Ligett, and L. Wasserman. Differential privacy with compression. In *2009 IEEE International Symposium on Information Theory*, pages 2718–2722. IEEE, 2009.