ONLINE DIFFERENTIALLY PRIVATE SYNTHETIC DATA GENERATION

YIYUN HE, ROMAN VERSHYNIN, AND YIZHE ZHU

Abstract

We present a polynomial-time algorithm for online differentially private synthetic data generation. 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 $O(t^{-1/d}\log(t))$ for $d \geq 2$ and $O(t^{-1}\log^{4.5}(t))$ for $d = 1$ in the 1-Wasserstein distance. This result generalizes the previous work on the continual release model for counting queries to include Lipschitz queries. Compared to the offline case, where the entire dataset is available at once [7, 36], our approach requires only an extra polylog factor in the accuracy bound.

1. INTRODUCTION

Differential privacy (DP) has emerged as a leading standard for safeguarding privacy in scenarios that involve the analysis of extensive data collections. It aims to protect the information of individual participants within datasets from disclosure. At its core, an algorithm is differentially private if it can produce consistently randomized outcomes for nearly identical datasets. This approach to privacy preservation is gaining traction across various sectors, notably in the implementation of the 2020 US Census [1, 34, 33] and among technology companies [17, 22]. The scope of differential privacy extends to a wide range of data science applications, including statistical query [48], regression [15, 54], parameter estimation [21], and stochastic gradient descent [52].

Existing research focuses on developing specialized algorithms for specific tasks constrained by predefined queries. This necessitates a significant level of expertise and often requires the modification of existing algorithms. Addressing these challenges, a promising approach is to create a synthetic dataset that approximates the statistical properties of the original dataset while ensuring differential privacy [32, 4]. This approach facilitates subsequent analytical tasks on the synthetic dataset without additional privacy risks.

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 or collecting location data from vehicles [44]. In these contexts, generating online synthetic data that adheres to differential privacy standards poses significant challenges [11, 46].

1.1. Continual release model.

One popular model in online differential privacy is the *Continual Release Model*, first studied in [23, 14]. 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 [23, 14] was for Boolean data streams. At time t, a Boolean sequence $x_1, \ldots, x_t \in \{0,1\}$ is available, and DP 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 DP algorithm outputs an infinite sequence.

In the online setting, repeating offline DP-counting algorithms would require an increasing privacy budget over time due to the composition property of differential privacy [26], thus not being feasible.

A seminal contribution of [23, 14] is the Binary Mechanism, which achieves polylog(t) error while maintaining ε-differential privacy for a finite time horizon T. Additionally, [24] improved the accuracy of the Binary Mechanism to $O\left(\log(T)+\log^2(n)\right)$ 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 [24] is optimal and matches the $\Omega(\log T)$ lower bound in [23] for online DP-count release.

The Binary Mechanism serves as a foundational element for many online private optimization problems [31, 43]. Various methods to enhance the Binary Mechanism in different settings have been studied in [14, 24, 51, 29, 37, 38]. Besides counting tasks, DP algorithms for online data have also been discussed for mean estimation [30], moment statistics [50, 27], graph statistics [53], online convex programming [41], decaying sums [9], user stream processing [16], and histograms [13]. Utilizing offline DP algorithms as black boxes, [18] provided a general technique to adapt them to online DP algorithms with utility guarantees.

1.2. **Differentially private synthetic data.** Among all DP algorithms, DP synthetic data generation [2, 32, 5, 42, 4, 6, 60, 28, 47] excels in flexibility because it allows for a wide range of downstream tasks to be performed without incurring additional privacy budgets. However, the computationally efficient construction of DP synthetic data with utility guarantees remains challenging. [56] shows that it is impossible to use poly(d) samples to generate d-dimensional DP synthetic data that approximates all 2-way marginals in polynomial time.

Most results for private synthetic data are concerned with counting queries, range queries, or k-dimensional marginals [32, 56, 5, 57, 25, 55, 8], and various metrics have been used to evaluate the utility of DP synthetic data [6, 61, 3].

A recent line of work [7, 20, 36, 35, 19] provides utility guarantees with respect to the Wasserstein distance for DP synthetic data. Using the Kantorovich-Rubinstein duality [58], the 1-Wasserstein distance accuracy bound ensures that all Lipschitz statistics are uniformly preserved. Given that numerous machine learning algorithms are Lipschitz [59, 45, 10, 49], one can expect similar outcomes for the original and synthetic data.

1.3. **Main results.** We consider the problem of generating DP-synthetic data under the continual release model beyond the Boolean data setting considered in [23, 14]. The data stream comes from the hypercube $[0,1]^d$ 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 1.1 (Online differentially private synthetic data)**.** *There exists an ε-differentially private algorithm such that, for any data stream $x_1,\dots,x_t,\cdots\in[0,1]^d$, at any time t, transforms the first t points $\mathcal{X}_t=\{x_1,\dots,x_t\}$ into t points $\mathcal{Y}_t\subset[0,1]^d$, with the following accuracy bound:*

$$\mathbb{E}W_1(\mu_{\mathcal{X}_t},\mu_{\mathcal{Y}_t})\lesssim\begin{cases}\varepsilon^{-1}\log(t)\cdot t^{-\frac{1}{d}}, & d\geq 2,\\ (\varepsilon t)^{-1}\log^{4.5}(t), & d=1,\end{cases}\tag{1.1}$$

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.

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

The utility guarantee in Theorem 1.1 is optimal up to a $\log t$ factor for $d\geq 2$ and $\mathrm{polylog}(t)$ 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 $[0,1]^d$,

$d \geq 2$, [36] proposed an algorithm with utility bound $O(n^{-1/d})$, which matches the minimax lower bound proved in [7].

We prove Theorem 1.1 by analyzing two different algorithms for $d \geq 2$ and $d = 1$, respectively. In our main Algorithm 1 for $d \geq 2$, 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 1.

A key ingredient in our work is the development of a special *Inhomogeneous Sparse Counting Algorithm* (Algorithm 2) 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 [7], 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.

Comparison to previous results. Our work generalizes the framework of DP-counting queries for Boolean data in the continual release model [23, 14] to online synthetic data generation in a metric space. One of the subroutines, *Inhomogeneous Sparse Counting* (Algorithm 2), is a generalization of the sparse counting mechanism from [24] with inhomogeneous noise according to the online hierarchical partition structure we introduce. The Binary Mechanism in [23, 24] is designed only for a finite time horizon, and a modified Hybrid Mechanism was developed for the infinite case in [14]. Our Algorithm 1 works for data streams with an infinite time horizon.

In terms of online DP synthetic data generation, [11] studied online DP synthetic data with prefixed counting queries, and [46] considered online DP-synthetic data for spatial datasets. 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. [12] established a separation for the number of offline, online, and adaptive queries subject to differential privacy. For the continual release model, [40] 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 [23] between online and offline counting tasks.

Our Theorem 1.1 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 [23] 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 [23] 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 1.1 is tight in terms of the dependence on t.

Organization of the paper. The rest of the paper is organized as follows. Section 2 introduces several notations and definitions. Section 3 details the online DP-synthetic data generation algorithm. In Section 4, we demonstrate that Algorithm 1 is differentially private with the accuracy bound stated in Theorem 1.1 for $d \geq 2$. Appendix A presents a modified version of the sparse counting algorithm

from [25], along with the corresponding privacy and accuracy analysis in our setting. Proofs of Lemmas and Theorem 1.1 for $d = 1$ are provided in Appendix B. The analysis of the time complexity of Theorem 1.1 is included in Appendix C.

2. Preliminaries

2.1. **Differential privacy.** We use the following definitions of differential privacy and neighboring datasets from [26].

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 data $\mathcal{X}, \mathcal{X}'$ and any subset S,*

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

2.2. **Online synthetic data generation.** We consider DP algorithms to release synthetic data in an online fashion when a data stream arrives. In our setting, a dataset is an infinite sequence

$$\mathcal{X} = (x_1, \dots, x_t, \dots),$$

where each $x_t \in [0,1]^d$ arrives at time $t \in \mathbb{Z}_+$. Two datasets $\mathcal{X}, \mathcal{X}'$ 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

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

Note that elements in $\mathcal{Y}_t$ do not need to be included in $\mathcal{Y}_{t+1}$.

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.

2.3. **Wasserstein distance.** Consider two probability measures μ, ν in a metric space (Ω, ρ). The 1-Wasserstein distance for more details) between them is defined as

$$W_1(\mu, \nu) := \inf_{\gamma \in \Gamma(\mu,\nu)} \int_{\Omega \times \Omega} \rho(x, y) \mathrm{d}\gamma(x, y),$$

where $\gamma(\mu, \nu)$ is the set of all couplings of μ and ν.

To quantify the utility of an online DP-synthetic data algorithm $\mathcal{A}$, we compare the statistical properties of the synthetic data sets $\mathcal{Y}_1, \dots, \mathcal{Y}_t$ with the true datasets $\mathcal{X}_1, \dots, \mathcal{X}_t$ for all $t \in \mathbb{Z}_+$. We identify each data set with an empirical measure and define

$$\mu_{\mathcal{X}_t} = \frac{1}{t} \sum_{i=1}^{t} \delta_{x_i}, \quad \mu_{\mathcal{Y}_t} = \frac{1}{t} \sum_{i=1}^{t} \delta_{y_{i,t}}.$$

We would like to have the synthetic data stream $(\mathcal{Y}_1, \dots, \mathcal{Y}_t)$ to stay close to the data stream $(\mathcal{X}_1, \dots, \mathcal{X}_t)$ under 1-Wasserstein distance for all $t \in \mathbb{Z}_+$.

For two probability measures μ and ν on Ω, the Kantorovich-Rubinstein duality (see e.g., [58] for more details) gives:

$$W_1(\mu,\nu) = \sup_{f\in\mathcal{F}} \Big(\int_\Omega f \mathrm{d}\mu - \int_\Omega f \mathrm{d}\mu \Big), \tag{2.1}$$

where $\mathcal{F}$ denotes the function class of all 1-Lipschitz functions.

2.4. **Integer Laplacian distribution.** Since our method involves private counts of data points in different regions, we will use integer Laplacian noise to ensure they are integers. An *integer (or discrete) Laplacian distribution* [39] 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 $p_\sigma = \exp(-1/\sigma)$. A random variable $Z \sim \mathrm{Lap}_{\mathbb{Z}}(\sigma)$ is mean-zero and sub-exponential with variance $\mathrm{Var(Z)} \le 2\sigma^2$.

3. Online synthetic data

3.1. **Binary partition.** We follow the definition of binary hierarchical partition as described in [36]. A binary hierarchical partition of a set Ω of depth r is a family of subsets Ω_θ, indexed by $\theta \in \{0,1\}^{\le r}$,

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

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 $\{0,1\}^0$ consists of a single element $\emptyset$. When $\theta \in \{0,1\}^j$, we refer to j as the level of θ. When $\Omega = [0,1]^d$ equipped with the ℓ_∞-norm, a subregion Ω_θ with $\theta \in 0,1^j$ has a volume of 2^{-j} and $\mathrm{diam}(\Omega_\theta) \asymp 2^{-\lfloor j/d \rfloor}$.

Let $(\Omega_\theta)\theta \in 0,1^{\le r}$ represent a binary partition of Ω. Given a true data stream $(x_1,\ldots,x_t) \in \Omega^t$, the true count $n_\theta^{(t)}$ is the number of data points in the region Ω_θ at time t, i.e.,

$$n_\theta^{(t)} := \Big| \big\{ i \in [t] : \ x_i \in \Omega_\theta \big\} \Big|.$$

We can also 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.

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.

3.2. **Main algorithm.** We can now introduce our main algorithm for online differentially private synthetic data release, described formally in Algorithm 1. It can be summarized in 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 time $t_j = 2^j$. Note that any sub-region Ω_θ with $|\theta| = j$ in Algorithm 1 only exists from level j: before level $|\theta|$ it was not created and after time t_{j+1} it will be refined.

(2) For each existing region Ω_θ at time t, we output a perturbed count $N_\theta^{(t)}$ using a new *Inhomogeneous Sparse Counting Algorithm* described in Algorithm 2. 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

in every perturbation step. Privacy and accuracy guarantees for Algorithm 2 are given in Section 3.3.

(3) The noisy counts in the Perturbation step could be negative and inconsistent. We post-process them to ensure they are nonnegative, and the counts of subregions always add up to the region's count. The details of this step are given in Algorithm 3, see Section 3.4.

(4) 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 1 Online synthetic data

Input: Privacy budget ε. Infinite sequence $\{x_i\}_{i=1}^{\infty}$. For each time t, data points $(x_1, \dots, x_t)$ are available.

Initialization: Set $t = t_0 = 1$, $\Omega_\emptyset = \Omega$, and the depth of partition tree $r = 0$.

while $t \in \mathbb{N}$ **do**

if $t \geq 2^r$ **then** $r \leftarrow r + 1$.

(Binary Partition) Partition Ω_θ into $\Omega_{\theta 0}$ and $\Omega_{\theta 1}$ for every $|\theta| = r - 1$.

For every newly created Ω_θ, apply Algorithm 2 and start a subroutine denoted by $\mathcal{A}_\theta$ with starting level $r_0 = r$ and privacy parameters $\varepsilon_{j,r}, \varepsilon_{j,r+1}, \dots$.

end if

(Perturbation) For every Ω_θ, where $1 \leq |\theta| \leq r$, compute the noisy online count $N_\theta^{(t)}$ using subroutine $\mathcal{A}_\theta$ with one new data $\mathbf{1}_{\{x_t \in \Omega_\theta\}}$ in its input Boolean data stream.

(Consistency) Transform the perturbed counts $\{N_\theta^{(t)}\}_{|\theta|<r}$ 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 within each subregion Ω_θ where $|\theta| = r$.

Let $t \leftarrow t + 1$.

end while

3.3. **Noisy online count.** We now provide a detailed description of the Perturbation step in Algorithm 1. Our goal is to output a noisy count of the points $x_1, \dots, x_t$ 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 count release under continual observation for Boolean data studied in [23, 14, 24]. In particular, when the Boolean data stream is sparse, [24] improved the accuracy guarantee by a sparse counting algorithm. We include a description of the algorithm from [24] in Algorithm 4, see Appendix A.

Algorithm 2 is based on [24] and uses integer Laplacian noise with different variances in different time intervals. We now give several definitions to describe Algorithm 2:

- *Time level*: Starting from level 0, we say time t is at *level* j if $2^j \leq t < 2^{j+1}$. In Algorithm 2, we process the data stream level by level, where level j starts from the timestamp $t_j = 2^j$.
- *Starting level*: We set an additional input r_0 to indicate the level from which the output starts. More precisely, the output of Algorithm 2 start from time $t_{r_0} = 2^{r_0}$.

  We use $\widetilde{S}$ to store the private count from starting level r_0, and $\widetilde{S}$ does not include the count of data points arriving before the starting level r_0.

Algorithm 2 Inhomogeneous Sparse Counting

Input: Output starting level r_0 with default value 0. Boolean data sequence $\{X_t\}_{t=2^{r_0}}^{\infty}$. Noise parameters $\varepsilon_{r_0}, \varepsilon_{r_0+1}, \dots$.

Initialization: Set the finite private count $\widetilde{S} \leftarrow 0$, the current level $r \leftarrow r_0$.

while $r_0 \le r < \infty$ **do**

Counting subroutine: For $t \in [2^r, 2^{r+1})$, apply Algorithm 4 with time horizon 2^r and privacy parameter $\varepsilon_r/2$. Record the outputs $c_{2^r}, \dots, c_{2^{r+1}-1}$.

Output. Output $\widetilde{S} + c_t$ as the private count N_t at time t for $t \in [2^r, 2^{r+1})$.

Update

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

and start a new level with $r \leftarrow r+1$.

end while

- *Counting subroutine*: The subroutine Algorithm 4 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 it to count the number of 1's arriving in the time interval $[2^r, 2^{r+1})$, and output $N_t = \widetilde{S} + c_t$ as the noisy count up to time t for each $t \in [2^r, 2^{r+1})$.
- *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 $\sum_{t=2^r}^{2^{r+1}-1} X_t + \mathrm{Lap}_{\mathbb{Z}}(2/\varepsilon_r)$.

The following lemma is a privacy guarantee for Algorithm 2. We show Algorithm 2 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 3.1. *Let $\mathcal{A}$ be Algorithm 2. For two datasets $\mathcal{X}, \mathcal{X}'$ which differ on one data point at time $t \in [2^r, 2^{r+1})$, and for any measurable subset S in the range of $\mathcal{A}$, the following holds:*

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

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

Lemma 3.2 bounds the difference between a noisy count and the true count in Algorithm 2 for different time intervals.

Lemma 3.2. *For each time $t \in [2^r, 2^{r+1})$, let N_t be the output of the noisy count at time t in Algorithm 2. We have*

$$\mathbb{E}\,|N_t - S_t| \lesssim \sum_{i=1}^{2^{r_0}-1} X_i + \sum_{i=r_0}^{r-1} \frac{1}{\varepsilon_i} + \frac{\log t + \log^2 n_r}{\varepsilon_r},$$

where $S_t := \sum_{i=1}^{t} X_i$ is the true count at time t and $n_r := \sum_{i=2^r}^{2^{r+1}} X_i$.

3.4. **Consistency.** The consistency step is adapted from [36, Algorithm 3], as described in Algorithm 3. We first convert all negative noisy counts to 0 and then transform the entire sequence of noisy counts for the hierarchical partition into a consistent count, ensuring the counts from subregions add up to the count at the next level. In terms of the partition tree $\mathcal{T}$ described in Section 3.1, consistency means that the counts from any two nodes sharing the same parent will sum to the count

Algorithm 3 Consistency

Input: Integer sequence $(n'_\theta)_{\theta\in\{0,1\}^{\le r}}$ corresponding to the private count of each Ω_θ.
For the case $j=0$, set $m \leftarrow \max(n',0)$.
for $j=0,\dots,r-1$ **do**
  for $\theta\in\{0,1\}^j$ **do**
    Set the counts $n'_{\theta 0}\leftarrow\max(n'_{\theta 0},0)$, $n'_{\theta 1}\leftarrow\max(n'_{\theta 1},0)$.
    Transform the vector $(n'_{\theta 0},n'_{\theta 1})\in\mathbb{Z}_+^2$ into any vector $(m_{\theta 0},m_{\theta 1})\in\mathbb{Z}_+^2$ s.t.

$$m_{\theta 0}+m_{\theta 1}=m_\theta;\quad (m_{\theta 0}-n_{\theta 0})(m_{\theta 1}-n_{\theta 1})\ge 0$$

  end for
end for
Output: non-negative integers $(m_\theta)_{\theta\in\{0,1\}^{\le r}}$.

from the parent node. This step is crucial for Algorithm 1 to obtain a probability measure close to the empirical measure $\mu_{\mathcal{X}_t}$ in the Wasserstein distance.

4. Proof of Theorem 1.1 when $d\ge 2$

We now prove that when $d\ge 2$, Algorithm 1 satisfies the privacy and accuracy guarantee in Theorem 1.1. To complete our proof, in Algorithms 1 and 2, we choose privacy parameters

$$\varepsilon_{j,r}=C_1\varepsilon 2^{(j-r)(1-1/d)/2},\quad \text{where } C_1=\frac{1-2^{-(1-1/d)/2}}{2} \tag{4.1}$$

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

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

4.1. Privacy.

Proposition 4.1. *With the choice of privacy parameters as Equation (4.1) in Algorithm 2, we have Algorithm 1 is ε-differentially private.*

Proof. Since the privacy budget in Algorithm 1 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 $r=\lfloor\log_2 t\rfloor$. 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 $r+1,r+2,\dots$ 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}, $j\ge 0$. We consider the subroutine $\mathcal{A}_{\theta_j}$ in each of the regions Ω_{θ_j} for all $j\ge 0$. There are two cases:

(1) When $0 < j \le r$, Ω_{θ_j} counts the data X_t at time t. So by Lemma 3.1, we protect the privacy of X_t with parameter $\varepsilon_{j,r}$.
(2) When $j > r$, by the Initialization step in Algorithm 2, $\mathcal{A}_{\theta_j}$ and the private count $N_{\theta_j}^{(t)}$ no longer depends on the value of X_t.

By the parallel composition rule of differential privacy [26, Theorem 3.16] and taking a supremum over all possible t, we have Algorithm 1 is differentially private with parameter

$$\sup_{s\ge 0}\sum_{j=1}^{s}\varepsilon_{j,s} \le \frac{\varepsilon}{2}.$$

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 again. □

4.2. **Accuracy.** Now we consider the accuracy of the output in Wasserstein distance at time t with the corresponding level $r = \lfloor \log_2 t \rfloor$. We prove (1.1) below for $d \ge 2$.

Proof of (1.1). Let $\lambda_\theta^{(t)} = N_\theta^{(t)} - n_\theta^{(t)}$ be the counting noise of subregion Ω_θ at time t with $1 \le j = |\theta| \le r$. Note that such Ω_θ is created at time level j, which is also the starting level parameter in subroutine $\mathcal{A}_\theta$. By Lemma 3.2,

$$\mathbb{E}\left|\lambda_\theta^{(t)}\right| \lesssim \left|\{x_s \mid s < 2^j, x_s \in \Omega_\theta\}\right| + \sum_{i=j}^{r-1}\frac{1}{\varepsilon_{j,i}} + \frac{\log t + \log^2 n_j}{\varepsilon_{j,r}}. \tag{4.2}$$

For the Consistency step in Algorithm 3, we can follow the proof of [35, Theorem 4.2] and obtain

$$\mathbb{E}\, W_1(\mu_{\mathcal{X}_t}, \mu_{\mathcal{Y}_t}) \le \frac{1}{t}\left[\sum_{j=1}^{r-1}\sum_{\theta\in\{0,1\}^j}\mathbb{E}\left[\max\left(\left|\lambda_{\theta 0}^{(t)}\right|, \left|\lambda_{\theta 1}^{(t)}\right|\right)\right]\operatorname{diam}(\Omega_\theta)\right] + \delta, \tag{4.3}$$

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 $2^{|\theta|}$ many different subregions of such size.

Note that for fixed j, $\varepsilon_{j,i}$ decreases as i increases. From (4.3) and (4.2), we have

$$\begin{aligned}
\mathbb{E}\, W_1(\mu_{\mathcal{X}_t}, \mu_{\mathcal{Y}_t}) &\le \frac{1}{t}\sum_{j=1}^{r}\sum_{|\theta|=j}\mathbb{E}\left|\lambda_\theta^{(t)}\right| \cdot 2^{-j/d} + 2^{-r/d} \\
&\lesssim \frac{1}{t}\sum_{j=1}^{r}\sum_{|\theta|=j}\left(\left|\{x_s \mid s < 2^j, x_s \in \Omega_\theta\}\right| + \sum_{i=j}^{r-1}\frac{1}{\varepsilon_{j,i}}\right. \\
&\qquad \left. + \frac{\log t + \log^2\left(n_\theta^{(t)} + 1\right)}{\varepsilon_{j,r}}\right) \cdot 2^{-j/d} + 2^{-r/d} \\
&\lesssim \frac{1}{t}\sum_{j=1}^{r}\left(2^j + \sum_{|\theta|=j}\frac{\log t + \log^2\left(n_\theta^{(t)} + 1\right)}{\varepsilon_{j,r}}\right) \cdot 2^{-j/d} + 2^{-r/d}.
\end{aligned} \tag{4.4}$$

Since

$$\left(\log^2(x+e)\right)'' = \frac{2(1-\log(x+e))}{(x+e)^2} \le 0, \quad \text{if } x \ge 0,$$

the function $\log^2(x+e)$ is concave on $[0,+\infty)$. Therefore, for fixed j, we can apply Jensen's inequality when summing the $\log^2(n_\theta^{(t)}+1)$ terms over all $|\theta|=j$ and obtain

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

Substitute the result above into (4.4) and we have

$$\mathbb{E}\,W_1(\mu_{\mathcal{X}_t},\mu_{\mathcal{Y}_t}) \lesssim \frac{1}{t}\sum_{j=1}^{r}\left(1+\frac{1}{\varepsilon_{j,r}}\Big(\log t+\log^2\left(\frac{t}{2^j}+e\right)\Big)\right)2^{j(1-1/d)}+2^{-r/d}. \tag{4.5}$$

As $t\in[2^r,2^{r+1})$, we have $\log t\le r+1\lesssim r$ and

$$\log(t/2^j+e)\le\log(2^{r-j+1}+e)\lesssim r-j+1.$$

To attain ε-differentially privacy with ε, we can choose the privacy parameters to optimize the accuracy in Wasserstein distance. One of the nearly best choices is given in Equation (4.1). Therefore, we deduce that

$$\sum_{j=1}^{r}\frac{1}{\varepsilon_{j,r}}\Big(\log t+\log^2\left(\frac{t}{2^j}+1\right)\Big)2^{j(1-1/d)}\lesssim\frac{2^{r(1-1/d)/2}}{\varepsilon}\sum_{j=1}^{r}\Big(r+(r-j+1)^2\Big)2^{j(1-1/d)/2}. \tag{4.6}$$

When $d\ge 2$, the entry $2^{j(1-1/d)/2}$ would grow exponentially as j increases. Consider the following part in the sum:

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

where $\alpha=2^{\frac{1-1/d}{2}}\in[2^{1/4},\sqrt{2})$. With the detailed computation given in Appendix B.3, we have

$$S\lesssim\alpha^r, \tag{4.7}$$

which implies

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

Hence (4.6) is bounded by $O\left(\frac{r}{\varepsilon}\cdot 2^{r(1-1/d)}\right)$. Therefore, when $d\ge 2$, with (4.5) and (4.6) we have

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

This finishes the proof. □

Acknowledgements. R.V. is partially supported by NSF grant DMS-1954233, NSF grant DMS-2027299, U.S. Army grant 76649-CS, and NSF-Simons Research Collaborations on the Mathematical and Scientific Foundations of Deep Learning. Y.Z. is partially supported by NSF-Simons Research Collaborations on the Mathematical and Scientific Foundations of Deep Learning and the AMS-Simons Travel Grant.

REFERENCES

[1] John Abowd, Robert Ashmead, Garfinkel Simson, Daniel Kifer, Philip Leclerc, Ashwin Machanavajjhala, and William Sexton. Census topdown: Differentially private data, incremental schemas, and consistency with public knowledge. *US Census Bureau*, 2019.

[2] John M Abowd and Lars Vilhuber. How protective are synthetic data? In *International Conference on Privacy in Statistical Databases*, pages 239–246. Springer, 2008.

[3] Amir R Asadi and Po-Ling Loh. On the Gibbs exponential mechanism and private synthetic data generation. In *2023 IEEE International Symposium on Information Theory (ISIT)*, pages 2213–2218. IEEE, 2023.

[4] Steven M Bellovin, Preetam K Dutta, and Nathan Reitinger. Privacy and synthetic datasets. *Stan. Tech. L. Rev.*, 22:1, 2019.

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

[6] March Boedihardjo, Thomas Strohmer, and Roman Vershynin. Covariance's loss is privacy's gain: Computationally efficient, private and accurate synthetic data. *Foundations of Computational Mathematics*, pages 1–48, 2022.

[7] March Boedihardjo, Thomas Strohmer, and Roman Vershynin. Private measures, random walks, and synthetic data. *arXiv preprint arXiv:2204.09167*, 2022.

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

[9] Jean Bolot, Nadia Fawaz, Shanmugavelayutham Muthukrishnan, Aleksandar Nikolov, and Nina Taft. Private decayed predicate sums on streams. In *Proceedings of the 16th International Conference on Database Theory*, pages 284–295, 2013.

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

[11] Mark Bun, Marco Gaboardi, Marcel Neunhoeffer, and Wanrong Zhang. Continual release of differentially private synthetic data. *arXiv preprint arXiv:2306.07884*, 2023.

[12] Mark Bun, Thomas Steinke, and Jonathan 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.

[13] Adrian Rivera Cardoso and Ryan 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.

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

[15] Kamalika Chaudhuri and Claire Monteleoni. Privacy-preserving logistic regression. *Advances in neural information processing systems*, 21, 2008.

[16] Yan Chen, Ashwin Machanavajjhala, Michael Hay, and Gerome 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.

[17] Graham Cormode, Somesh Jha, Tejas Kulkarni, Ninghui Li, Divesh Srivastava, and Tianhao Wang. Privacy at scale: Local differential privacy in practice. In *Proceedings of the 2018 International Conference on Management of Data*, pages 1655–1658, 2018.

[18] Rachel Cummings, Sara Krehbiel, Kevin A Lai, and Uthaipon Tantipongpipat. Differential privacy for growing databases. *Advances in Neural Information Processing Systems*, 31, 2018.

[19] Konstantin Donhauser, Javier Abad, Neha Hulkund, and Fanny Yang. Privacy-preserving data release leveraging optimal transport and particle gradient descent. *arXiv preprint arXiv:2401.17823*, 2024.

[20] Konstantin Donhauser, Johan Lokna, Amartya Sanyal, March Boedihardjo, Robert Hönig, and Fanny Yang. Certified private data release for sparse lipschitz functions. *arXiv preprint arXiv:2302.09680*, 2023.

[21] John C Duchi, Michael I Jordan, and Martin J Wainwright. Minimax optimal procedures for locally private estimation. *Journal of the American Statistical Association*, 113(521):182–201, 2018.

[22] Cynthia Dwork, Nitin Kohli, and Deirdre Mulligan. Differential privacy in practice: Expose your epsilons! *Journal of Privacy and Confidentiality*, 9(2), 2019.

[23] Cynthia Dwork, Moni Naor, Toniann Pitassi, and Guy N Rothblum. Differential privacy under continual observation. In *Proceedings of the forty-second ACM symposium on Theory of computing*, pages 715–724, 2010.

[24] Cynthia Dwork, Moni Naor, Omer Reingold, and Guy 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.

[25] Cynthia Dwork, Aleksandar Nikolov, and Kunal Talwar. Efficient algorithms for privately releasing marginals via convex relaxations. *Discrete & Computational Geometry*, 53:650–673, 2015.

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

[27] Alessandro Epasto, Jieming Mao, Andres Munoz Medina, Vahab Mirrokni, Sergei Vassilvitskii, and Peilin Zhong. Differentially private continual releases of streaming frequency moment estimations. *arXiv preprint arXiv:2301.05605*, 2023.

[28] Chenglin Fan, Ping Li, and Xiaoyun Li. Private graph all-pairwise-shortest-path distance release with improved error rate. *Advances in Neural Information Processing Systems*, 35:17844–17856, 2022.

[29] Hendrik Fichtenberger, Monika Henzinger, and Jalaj Upadhyay. Constant matters: Fine-grained error bound on differentially private continual observation. In *International Conference on Machine Learning*, pages 10072–10092. PMLR, 2023.

[30] Anand Jerry George, Lekshmi Ramesh, Aditya Vikram Singh, and Himanshu Tyagi. Continual mean estimation under user-level privacy. *arXiv preprint arXiv:2212.09980*, 2022.

[31] Abhradeep Guha Thakurta and Adam Smith. (nearly) optimal algorithms for private online learning in full-information and bandit settings. *Advances in Neural Information Processing Systems*, 26, 2013.

[32] Moritz Hardt, Katrina Ligett, and Frank McSherry. A simple and practical algorithm for differentially private data release. *Advances in neural information processing systems*, 25, 2012.

[33] Mathew E Hauer and Alexis R Santos-Lozada. Differential privacy in the 2020 census will distort covid-19 rates. *Socius*, 7:2378023121994014, 2021.

[34] Michael B Hawes. Implementing differential privacy: Seven lessons from the 2020 United States Census. *Harvard Data Science Review*, 2(2), 2020.

[35] Yiyun He, Thomas Strohmer, Roman Vershynin, and Yizhe Zhu. Differentially private low-dimensional representation of high-dimensional data. *arXiv preprint arXiv:2305.17148*, 2023.

[36] Yiyun He, Roman Vershynin, and Yizhe Zhu. Algorithmically effective differentially private synthetic data. *arXiv preprint arXiv:2302.05552*, 2023.

[37] Monika Henzinger, Jalaj Upadhyay, and Sarvagya 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.

[38] Monika Henzinger, Jalaj Upadhyay, and Sarvagya 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.

[39] Seidu Inusah and Tomasz J Kozubowski. A discrete analogue of the laplace distribution. *Journal of statistical planning and inference*, 136(3):1090–1102, 2006.

[40] Palak Jain, Sofya Raskhodnikova, Satchit Sivakumar, and Adam Smith. The price of differential privacy under continual observation. *arXiv preprint arXiv:2112.00828*, 2021.

[41] Prateek Jain, Pravesh Kothari, and Abhradeep Thakurta. Differentially private online learning. In *Conference on Learning Theory*, pages 24–1. JMLR Workshop and Conference Proceedings, 2012.

[42] James Jordon, Jinsung Yoon, and Mihaela Van Der Schaar. PATE-GAN: Generating synthetic data with differential privacy guarantees. In *International conference on learning representations*, 2019.

[43] Peter Kairouz, Brendan McMahan, Shuang Song, Om Thakkar, Abhradeep Thakurta, and Zheng Xu. Practical and private (deep) learning without sampling or shuffling. In *International Conference on Machine Learning*, pages 5213–5225. PMLR, 2021.

[44] Jong Wook Kim, Kennedy Edemacu, Jong Seon Kim, Yon Dohn Chung, and Beakcheol Jang. A survey of differential privacy-based techniques and their applicability to location-based services. *Computers & Security*, 111:102464, 2021.

[45] Leonid V Kovalev. Lipschitz clustering in metric spaces. *The Journal of Geometric Analysis*, 32(7):188, 2022.

[46] Girish Kumar, Thomas Strohmer, and Roman Vershynin. An algorithm for streaming differentially private data. *arXiv preprint arXiv:2401.14577*, 2024.

[47] Jingcheng Liu, Jalaj Upadhyay, and Zongrui Zou. Optimal bounds on private graph approximation. In *Proceedings of the 2024 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1019–1049. SIAM, 2024.

[48] Ryan McKenna, Gerome Miklau, Michael Hay, and Ashwin Machanavajjhala. Optimizing error of high-dimensional statistical queries under differential privacy. *Proceedings of the VLDB Endowment*, 11(10), 2018.

[49] Laurent Meunier, Blaise J Delattre, Alexandre Araujo, and Alexandre Allauzen. A dynamical system perspective for Lipschitz neural networks. In *International Conference on Machine Learning*, pages 15484–15500. PMLR, 2022.

[50] Darakhshan Mir, Shan Muthukrishnan, Aleksandar Nikolov, and Rebecca 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.
[51] Yuan Qiu and Ke Yi. Differential privacy on dynamic data. *arXiv preprint arXiv:2209.01387*, 2022.
[52] Shuang Song, Kamalika Chaudhuri, and Anand D Sarwate. Stochastic gradient descent with differentially private updates. In *2013 IEEE global conference on signal and information processing*, pages 245–248. IEEE, 2013.
[53] Shuang Song, Susan Little, Sanjay Mehta, Staal Vinterbo, and Kamalika Chaudhuri. Differentially private continual release of graph statistics. *arXiv preprint arXiv:1809.02575*, 2018.
[54] Dong Su, Jianneng Cao, Ninghui Li, Elisa Bertino, and Hongxia Jin. Differentially private k-means clustering. In *Proceedings of the sixth ACM conference on data and application security and privacy*, pages 26–37, 2016.
[55] Justin Thaler, Jonathan Ullman, and Salil 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.
[56] Jonathan Ullman and Salil Vadhan. Pcps and the hardness of generating private synthetic data. In *Theory of Cryptography Conference*, pages 400–416. Springer, 2011.
[57] Giuseppe Vietri, Cedric Archambeau, Sergul Aydore, William Brown, Michael Kearns, Aaron Roth, Ankit Siva, Shuai Tang, and Steven Wu. Private synthetic data for multitask learning and marginal queries. In *Advances in Neural Information Processing Systems*, 2022.
[58] Cédric Villani. *Optimal transport: old and new*, volume 338. Springer, 2009.
[59] Ulrike von Luxburg and Olivier Bousquet. Distance-based classification with Lipschitz functions. *J. Mach. Learn. Res.*, 5(Jun):669–695, 2004.
[60] Bangzhou Xin, Yangyang Geng, Teng Hu, Sheng Chen, Wei Yang, Shaowei Wang, and Liusheng Huang. Federated synthetic data generation with differential privacy. *Neurocomputing*, 468:1–10, 2022.
[61] Yilin Yang, Kamil Adamczewski, Danica J Sutherland, Xiaoxiao Li, and Mijung Park. Differentially private neural tangent kernels for privacy-preserving data generation. *arXiv preprint arXiv:2303.01687*, 2023.

APPENDIX A. SPARSE COUNTING ALGORITHM FROM [24]

This section includes the online counting algorithm in [24], which is ε-DP with an optimal accuracy error $O(\log T)$ when the data stream is sparse. The detailed algorithm is given in Algorithm 4.

Algorithm 4 Sparse counting, finite time horizon [24].

Input: Time range $T < \infty$, Boolean data sequence $\{X_i\}_{i=1}^T$ with n many 1's. 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 $\widetilde{S} = 0$ denote the private count of the previous segments.
subroutine Start an online counting subroutine $\mathcal{A}_{\text{sub}}$ with parameter $\varepsilon/2$ and input to be determined later.

for segment j **do**
  Set starting time $t_j = t + 1$ of the current segment.
  Set the segment count $S_j = 0$ and the private threshold $\widetilde{T}_j = T_0 + \text{Lap}_{\mathbb{Z}}(2/\varepsilon)$

  while $S_j + \text{Lap}_{\mathbb{Z}}(2/\varepsilon) \le \widetilde{T}_j$ and $t \le T$ **do**
    Set $t = t + 1$, $S_j = S_j + X_t$.
    Output. Output the same $\widetilde{S}$ for all t in this segment.
  end while
  Run $\mathcal{A}_{\text{sub}}$ with one newly coming data S_j. Add the output to $\widetilde{S}$, the private count of the previous segments.
  Set $j \leftarrow j + 1$
end for

The algorithm has various choices of the online counting subroutine $\mathcal{A}_{\text{sub}}$. One choice is the *Hybrid counting mechanism* proposed in [14], which ensures differential privacy for an infinite input data stream.

Lemma A.1 ([14])**.** *The* Hybrid counting mechanism *is ε-differentially private for an infinite time horizon. And for any time $t > 0$ and $\beta > 0$, with probability at least $1 - \beta$, the counting error is $\frac{C}{\varepsilon} \cdot \log^{1.5} t \cdot \log \frac{1}{\beta}$.*

Although such error bound is in probability, we can easily transform it into a similar expectation bound by a simple equality in probability. 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

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

After doing 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

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

Therefore, we have the expectation error bound

$$\mathbb{E}\,\text{error}_t = \frac{C}{\varepsilon} \log^{1.5} t \tag{A.1}$$

With the guarantee of the subroutine, [23] shows that the sparse counting Algorithm 4 indeed improves the counting error. As the value of the partition threshold T_0 is related to an extra parameter, the confidence probability β, in [23], we will prove a similar expectation bound with the new partition threshold T_0 in Algorithm 4.

Lemma A.2. *Algorithm 4 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*

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

Proof. The privacy part follows from the original proof in [23]. 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 $\text{Lap}(2/\varepsilon)$ random variables in total. Therefore, by a simple union bound argument, with probability $1/T$, their magnitudes are uniformly bounded with bound $B = \frac{2}{\varepsilon}(2 \log T + \log 2)$. Conditioning on this event, we know $S_j + \text{Lap}_{\mathbb{Z}}(2/\varepsilon) > \widetilde{T}_j$ 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 $\text{Lap}(2/\varepsilon)$ random variables share the uniform bound B, the counting error consists of two parts: 1. the error from the online counting subroutine $\mathcal{A}_{\text{sub}}$ 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 (A.1), 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 $\mathcal{A}_{\text{sub}}$ 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

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

□

APPENDIX B. ADDITIONAL PROOFS

B.1. **Proof of Lemma 3.1.**

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 third assertion holds.

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

(1) When $\widetilde{S}$ first counts X_t privately by 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 [26], we know for the data at time t, the algorithm is ε_r differentially private. □

B.2. **Proof of Lemma 3.2.**

Proof. The accuracy part also follows from the composition results of the subroutine and Laplacian mechanism. At time $t \in [2^r, 2^{r+1})$, by the result of sparse online counting Lemma A.2, we know the error of count c_t at time t has bound

$$\mathbb{E}\left|c_t-\sum_{i=t_r}^{t} X_i\right| \lesssim \frac{\log t+\log^2 n_r}{\varepsilon_r}.$$

And by the Laplacian mechanism, we know the count $\widetilde{S}$ has initialization error $O(1/\varepsilon')$ as well as the accumulating error from each level (starting from r_0), namely

$$\mathbb{E}\left|\widetilde{S}-\sum_{i=2^{r_0}}^{2^r-1} X_i\right| \lesssim \sum_{i=r_0}^{r-1}\frac{1}{\varepsilon_i}.$$

Therefore, considering that $\widetilde{S}$ ignored the the data before level $r_1 = \lfloor r_0(1-1/d) \rfloor$, we deduce that

$$\begin{aligned}\mathbb{E}|N_t - S_t| &= \mathbb{E}\left|\widetilde{S} + c_t - \sum_{i=1}^{t} X_i\right| \\ &\le \sum_{i=1}^{2^{r_1}-1} X_i + \mathbb{E}\left|\widetilde{S} - \sum_{i=2^{r_1}}^{2^r-1} X_i\right| + \mathbb{E}\left|c_t - \sum_{i=2^r}^{t} X_i\right| \\ &\lesssim \sum_{i=1}^{2^{r_1}-1} X_i + \sum_{i=0}^{r-1} \frac{1}{\varepsilon_i} + \frac{\log t + \log^2 n_r}{\varepsilon_r}.\end{aligned}$$

□

B.3. **Proof of (4.7).** We prove the following lemma:

Lemma B.1. *Suppose $\alpha > 1$ is a constant and r is a positive integer, then*

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

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

Here, the asymptotic results are with respect to r.

Proof. We can do the following trick

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

For the sum with quadratic entries, 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 the first result of summing with linear entries, we have

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

□

B.4. **Proof of Theorem 1.1 for $d = 1$.** Note that Algorithm 1 is built on Algorithm 2, which only works for $d \geq 2$ due to the choice of the privacy parameters in (4.1).

In this section, we present a simpler algorithm for $d = 1$, which can also be generalized for $d \geq 2$ with a weaker utility bound and prove its privacy.

To adjust to the case $d = 1$, the main structure of the online synthetic data algorithm remains the same, with the only modification on the choice of the subroutine $\mathcal{A}_\theta$ and its parameters. When $d = 1$, we can use any private binary counting mechanism as a subroutine $\mathcal{A}_\theta$. For example, a good choice is *Hybrid Mechanism* from [14, Corollary 4.8]. We included its privacy and accuracy result in Lemma A.1.

Proposition B.2. *In Algorithm 1, by changing $\mathcal{A}_\theta$ to* Hybrid Mechanism *with privacy parameter*

$$\frac{3}{\pi^2} \cdot \frac{\varepsilon}{(|\theta| + 1)^2}$$

and the same starting input Boolean data stream $\{\mathbf{1}_{\{x_i \in \Omega_\theta\}}\}_{i=1}^{t-1}$, the modified version of Algorithm 1 is ε-differentially private.

Proof. 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. Similar to the proof to Proposition 4.1, there is exactly one sub-region containing x_t on each level, namely $\Omega_{\theta_r} \subset \Omega_{\theta_{r-1}} \subset \cdots \subset \Omega$. Also, in the future there will also be $\Omega_{\theta_r} \supset \Omega_{\theta_{r+1}} \supset \Omega_{\theta_{r+2}} \supset \ldots$. Similarly we can also obtain an infinite sequence $\Omega \supset \Omega_{\theta_1'} \supset \Omega_{\theta_2'} \supset \cdots$ containing x_t'.

By parallel composition property of differential privacy [26], consider the influenced subroutine $\{\mathcal{A}_{\theta_j}\}_{j=0}^{\infty}$, we know the algorithm is differentially private with parameter ε due to the following identity:

$$\frac{\varepsilon}{2} = \sum_{j=0}^{\infty} \frac{3}{\pi^2} \cdot \frac{\varepsilon}{(j+1)^2}.$$

□

Now, we are ready to prove the utility bound in Theorem 1.1 for $d = 1$.

Proof of Theorem 1.1 for $d = 1$. We can next show its accuracy in Wasserstein distance with the online synthetic data algorithm defined in Proposition B.2. By Lemma A.1 and our discussion, at every time t, we have expectation bound

$$\mathbb{E}\left|N_\theta^{(t)} - n_\theta^{(t)}\right| \lesssim \frac{(|\theta| + 1)^2}{\varepsilon} \log^{1.5} t.$$

Note that $n_\theta^{(t)}$ is the true count of Ω_θ at time t, while $N_\theta^{(t)}$ is the private count of Ω_θ at time t and also the output of $\mathcal{A}_\theta$ at time t.

Define $\lambda_\theta^{(t)} = N_\theta^{(t)} - n_\theta^{(t)}$. Again, similar to the proof of the $d \geq 2$ case, we can apply the result in [36] and get Equation (4.3). For time $t \in [2^r, 2^{r+1})$, we have

$$\begin{aligned}
\mathbb{E}\, W_1(\mu_{\mathcal{X}_t}, \mu_{\mathcal{Y}_t}) &\leq \frac{1}{t}\left[\sum_{j=0}^{r-1}\sum_{\theta\in\{0,1\}^j} \mathbb{E}\left[\max\left(\left|\lambda_{\theta 0}^{(t)}\right|, \left|\lambda_{\theta 1}^{(t)}\right|\right)\right]\operatorname{diam}(\Omega_\theta)\right] + \delta, \\
&\leq \frac{1}{t}\sum_{j=1}^{r}\sum_{|\theta|=j}\mathbb{E}\left|\lambda_\theta^{(t)}\right| \cdot 2^{-j/d} + 2^{-r/d} \\
&\lesssim \frac{1}{t}\sum_{j=1}^{r} 2^j \cdot \frac{(j+1)^2}{\varepsilon}\log^{1.5} t \cdot 2^{-j/d} + 2^{-r/d} \\
&= \frac{\log^{1.5} t}{\varepsilon t}\sum_{j=1}^{r}(j+1)^2 2^{j(1-1/d)}.
\end{aligned}$$

For the particular case $d = 1$, we have $\sum_{j=1}^{r}(j+1)^2 \asymp r^3$. Hence, we have the accuracy bound

$$\mathbb{E}\, W_1(\mu_{\mathcal{X}_t}, \mu_{\mathcal{Y}_t}) \lesssim \frac{\log^{4.5} t}{\varepsilon n}.$$

This finishes the proof. □

Remark B.3. The deduction above works for general d. Therefore, when $d \geq 2$, the $2^{j(1-1/d)}$ term grows exponentially and $\sum_{j=0}^{r}(j+1)^2 2^{j(1-1/d)} \asymp r^2 2^{r(1-1/d)}$. With the simpler algorithm discussed in this section, we will obtain a slightly weaker accuracy bound

$$\mathbb{E}\, W_1(\mu_{\mathcal{X}_t}, \mu_{\mathcal{Y}_t}) \lesssim \frac{\log^{1.5} t}{\varepsilon n} \cdot r^2 2^{r(1-1/d)} \lesssim \frac{\log^{3.5} t}{\varepsilon} t^{-1/d}.$$

APPENDIX C. TIME COMPLEXITY

Due to the online setting, let us first consider the running time for the algorithms to output after the input data arrives at time t.

The Hybrid Counting Mechanism in [14] 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 4, as it checks the partition threshold with $O(1)$ time and runs Hybrid Counting Mechanism as a subroutine.

As for Algorithm 2, *Inhomogenous Sparse Counting*, with starting level r_0, it is connected by $O(\log t)$ many implementations of Algorithm 4, each of time horizon $t_{r_0} = 2^{r_0}, t_{r_0+1} = 2^{r_0+1}, \ldots$. Furthermore, for a given t, only one such subroutine is active, so the time complexity is also $O(\log t)$.

For main Algorithm 1, there is another variable d, the data dimension. For each fixed time t in Algorithm 1, there are $O(t)$ many sub-regions Ω_θ in the binary partition tree of $[0,1]^d$, hence the partition step has time complexity $O(dt)$. Also, one new data point comes at time t for all subroutines $\mathcal{A}_\theta$'s, which in total have running time $O(t \log t)$ by the result above for Algorithm 2. Finally, the time complexity is $O(t)$ for the consistency step and $O(dt)$ for the outputs. Therefore, the whole Algorithm 1 has time complexity $O(dt + t\log t)$ to output at time t.

When $d = 1$, we use the Hybrid Counting Mechanism directly, and the same time complexity holds.

DEPARTMENT OF MATHEMATICS, UNIVERSITY OF CALIFORNIA IRVINE, IRVINE, CA 92697
Email address: yiyunh4@uci.edu

DEPARTMENT OF MATHEMATICS, UNIVERSITY OF CALIFORNIA IRVINE, IRVINE, CA 92697
Email address: rvershyn@uci.edu

DEPARTMENT OF MATHEMATICS, UNIVERSITY OF CALIFORNIA IRVINE, IRVINE, CA 92697
Email address: yizhe.zhu@uci.edu