Differentially private synthetic data release from hybrid partitioned data

Gang Zhang
Beijing Institute of Technology

Ruinan Li
Beijing Institute of Technology

Zhongxiang Lei
Beijing Institute of Technology

Yuanchi Ma
Beijing Institute of Technology

Jinyan Liu
Beijing Institute of Technology

ABSTRACT

The integration and release of distributed data from multi-party with differentially private protection can provide robust support for contemporary data-driven research. Previous works focus on the setting of partitioned databases by a horizontally or a vertically manner, however, the realistic scenes can be more complexed. We firstly consider the private integration from a high-dimensional hybrid partitioning that combines both horizontally and vertically partitioned databases. In such context, each party only holds partial information about the complete dataset and its attributes. With the assistance of a semi-trusted curator, the parties collectively generate a synthetic data while achieving private protection for local dataset. In this paper, we present a novel exploration for releasing high-dimensional hybrid partitioned data, and propose RandDPGen to generate differentially private synthetic dataset. Specifically, we introduce distributed marginal measurement technique to address the challenge of performing noisy measurements on hybrid partitioned datasets, and employ a graphical-model-based estimation algorithm to mitigate computational difficulties inherent in high-dimensional data. These methods are then integrated into a select-measure-generate paradigm to generate private synthetic data, enabling informed decision-making and delivering high-quality services. We theoretically prove the privacy guarantee of proposed approach, and quantify uncertainty for the synthetic data with high probability. Furthermore, linear and non-linear experiments on two real datasets demonstrate the feasibility and desirable data utility of our approach.

1 INTRODUCTION

With the rapid advancement of data-driven research, particularly the advent of large language models, such as the GPT series and BERT, which have brought unprecedented convenience and groundbreaking progress to numerous disciplines[9, 15]. However, the subsequent issues of data privacy protection and data scarcity have significantly hindered the development of data-driven research[27]. Private synthetic data effectively addresses these limitations by providing virtual data that simulates the statistical characteristics of real-world data. Meanwhile, ensuring high-quality synthetic data generation while maintaining data privacy has become a key challenge.

Differential Privacy (DP)[7] as a rigorous privacy protection framework, provides a crucial trade-off between data utility and individual privacy in synthetic data generation. In recent years, the emergence of numerous variants of differential privacy has provided researchers with a variety of options [8, 23, 30]. Especially, [4] propose zero Concentrated Differential Privacy (zCDP) for the composition properties in more complex algorithms and data analysis tasks. It can provide smaller standard deviation in noise mechanism.

Based on above research, researchers first explore the generation and release of one single dataset with differential privacy. Hardt et al. introduced the select-measure-generate paradigm, combining exponential mechanisms with multiplicative weighting to privately generate synthetic data[11], marking a groundbreaking advancement for releasing a single dataset. Building upon these works, more sophisticated techniques[2, 18–20] have been developed to enhance the utility of synthetic data. Subsequently, this work is extended to the distributed setting[6, 28], i.e., the dataset is partitioned into several parts, and each party holds one. To delineate this scenario more precisely, we categorize the distributed data into two distinct holding configurations: horizontal (or individual-based) holding and vertical (or attribute-based) holding. For horizontal holding, such entries of a dataset are possessed by different parties as if the data were horizontally partitioned among multiple parties. For vertical holding, different attributes of a dataset are possessed by different parties as if the data were vertically partitioned among multiple parties. Recently, there have been some pioneering works in releasing horizontally partitioned datasets[1, 6, 12] and vertically partitioned datasets [24, 28].

However, the two scenarios mentioned above, which respectively assume that the parties hold all individual entities or attributes of the data, are seldom encountered in real-world applications. More than often, a party only holds portions of the complete set of data entries and attributes, as if the data is partitioned among multiple parties in a hybrid way that includes horizontal and vertical partitioning. In this paper, we first explore the hybrid partitioned data, aiming to generate its private synthetic data and advance the development of the data-driven research. The release of summary data from the US Decennial Census can be framed as a real-world example. In this context, the U.S. Census Bureau aggregates datasets from each state and release the synthetic data, including individual attributes such as gender, age, income, and other related information. However, each state retains possession of its own data, with various institutions overseeing different individual attributes.

The general model of the above scene is that the distributed parties (i.e., the data owners) and a semi-trusted curator, they correctly follow a pre-defined protocol to perform computations, but may use any intermediate results to infer sensitive information of

other parties. We adopt the select-measure-generate paradigm to generate differentially private synthetic data for the hybrid partitioned data in this scene, but two challenges remain to be addressed: 1) Statistical measurement on the distributed datasets: The select-measure-generate paradigm typically generate the synthetic data by measuring marginal distributions. However, given the semi-trusted setting, measuring marginal distributions on hybrid partitioned data under differentially private protection is intractable. 2) The computational burdens caused by high-dimensional data: Synthetic data driven by data-driven research often involves high-dimensional data with numerous attributes, which typically imposes a significant computational burden when constructing distributions.

To solve these challenges, we propose RandDPGen, a differentially private synthetic data generation algorithm designed for high-dimensional hybrid partitioned data. Specifically, we present distributed marginal measurement technique, which achieves privacy-preserving marginal measurements of distributed data by dexterously assigning sub-query to each local dataset. Then, we adopt the graphical models to efficiently estimate the high-dimensional distribution when the measurements are over low-dimensional marginals, which avoids the computational load caused by high-dimensional data. Finally, we integrate them into the select-measure-generate paradigm to propose RandDPGen for releasing private synthetic data. We fine-tune multiple parts of RandDPGen to improve its performance in generating private synthetic data. We summarize our contributions below:

- We propose and explored a significant issue: generating private synthetic data from hybrid partitioned datasets, representing the first comprehensive investigation of this critical challenge.
- We propose RandDPGen, a method for generating differentially private synthetic data from hybrid partitioned data. This method leverages distributed marginal measurements and graphical model to address the challenges of statistical measurements on distributed data and the computational burdens associated with high-dimensional data.
- We theoretically prove the private guarantee of our method, and derive the upper bound $O\left(\sqrt{\frac{KS^2}{\pi\rho_t}} + \frac{\triangle t}{\sqrt{\rho_t}}\left(\log|W|+1\right)\right)$ of its practical uncertainty with high probability.
- We evaluate the discrepancy between synthetic data and real-world datasets (Big5 and ADULT) across linear and non-linear tasks, demonstrating the feasibility of RandDPGen and its desirable data utility with a KL divergence score of 0.083 under $\epsilon = 0.1$ and 16 parties.

2 RELATED WORK

In recent years, synthetic data generation has emerged as a powerful tool for addressing privacy concerns while enabling data-driven research and machine learning applications. However, the generation and utilization of synthetic data in different data partitioning scenarios have unique challenges and requirements. In this section, we analyze two distinct yet interrelated aspects of private synthetic data generation: (1) generation for horizontally partitioned data, which involves creating synthetic data that aligns with distributed systems and enables cross-party collaboration; and (2) generation for vertically partitioned data, where synthetic data respects hierarchical relationships between attributes and ensures privacy preservation in feature-specific learning scenarios. By systematically analyzing these aspects, this work aims to address a more complex scenario that generating synthetic data for hybrid partitioned data.

Private synthetic data for horizontally partitioned data. There are a number of studies focus on generating private synthetic data on the horizontally partitioned data. These researches originated from the anonymous sharing of distributed data. Jurczyk and Xiong[14] develop l-site-diversity and a distributed anonymization protocol for building a virtual anonymized database while maintaining privacy constraints. Mohammed et al.[26] propose LKC-privacy, a new privacy model, along with two anonymization algorithms for addressing the private problem in centralized and distributed scenarios. Goryczka et al.[10] address collaborative anonymization of horizontally partitioned data, proposing m-privacy to guard against up to m colluding providers, along with efficient algorithms for verification and provider-aware anonymization. Different from these works, Alhadidi et al.[1] consider to release horizontally partitioned high-dimensional data under the protection of differential privacy. Hong et al.[12] achieve the trade-off between privacy and utility in collaborative search log by a sanitization framework. However, these works are constrained by the number of participating parties and the generalization capability of downstream tasks. Cheng et al.[6] present a differentially private sequential update of Bayesian network (DP-SUBN), and solve these challenges in horizontal Settings through this means.

Private synthetic data for vertically data. Similarly, a few researchers have focused on vertical settings. Jiang and Clifton[13] proposes a two-party framework for generating k-anonymous data from vertically partitioned sources, enabling multi-party data sharing while meeting privacy requirements. In order to break through the limitation of the two-party setting and the assumption of semi-honest behavior, Mohammed et al.[25] develop k-anonymity privacy model and game-theoretic algorithms in the field of multi-party data aggregation. In the exploration of differential privacy, Mohammed et al.[24] also propose DistDiffGen for differentially private data release between two parties with vertically partitioned data in the semi-honest adversary model. They accomplish this objective by creating a two-party protocol for the exponential mechanism and a differentially private data release algorithm that adheres to the definition of secure multiparty computation. Based on these works, Tang et al.[28] propose a differentially private latent tree (DPLT) method that addresses the issue of decreasing data utility with increasing data dimensions and limitations on the number of participants. In spite of it is widespread in practical applications, to our knowledge, no work has focused on releasing hybrid partitioned data that is held by multiple parties across both horizontal and vertical partitions.

3 PRELIMINARIES

In this section, we introduce the basic knowledge we used in this paper and the problem formulations.

3.1 Notations

Data. Let $\mathcal{X} = \{x_1, x_2, \cdots, x_d\}$ denote a finite d-dimensional data domain. A dataset D consisting of R records is denoted as $D \in \mathcal{X}^R$. The domain of possible values for x_i is denoted by Ω_i, which we assume has size $|\Omega| = n_i$. The all domain of possible values for any record x is thus $\Omega = \Omega_1 \times \cdots \times \Omega_d$ which has size $\prod_i n_i = n$. We denote the set of all possible datasets by $\mathcal{D}$, which is equal to $\bigcup_{R=0}^{\infty} \Omega^R$.

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

Definition 3.1. *(Mrginals). Let $r \in [d]$ be a subset of attributes, $\Omega_r = \prod_{i \in r} \Omega_i$, $n_r = |\Omega_r|$, and $x_r = (x_i)_{i \in r}$. The marginal on r is a vector $\mu \in \mathbb{R}^{n_r}$, indexed by domain lelments $t \in \Omega_r$, such that each entry is a count, i.e., $\mu[t] = \sum_{x \in D} \mathbb{1}[x_r = t]$. We let $M_r : \mathcal{D} \to \mathbb{R}^{n_r}$ denote the function that computes the marginal on r, i.e., $\mu = M_r(D)$.*

In this paper, we utilize the term marginal query to refer to the function M_r, and *marginal* to represent the vector of counts, denoted as $\mu = M_r(D)$. For simplicity, we occasionally refer to the attribute subset r as a marginal query as well. A k-way marginal query implies that $|r| = k$.

Workload. A workload W is a collection of queries that the synthetic data is expected to preserve well, and also is the measurement for evaluating the utility of different mechanisms. Our goal is to take a workload as input of our mechanisms and adapt intelligently to its queries, generating synthetic data that is specifically tailored to the queries of interest. In this paper, we construct the workload W with a collection of weighted marginal queries. The measurement is stated in Definition 3.2.

Definition 3.2. *(Workload Error). A workload W consists of a list of marginal queries $r_1, \ldots, r_k$ where $r_i \subseteq [d]$, together with associated weights $c_i \geq 0$. The error of a synthetic dataset $\hat{D}$ is defined as:*

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

Furthermore, we introduce the metrics for evaluating the differences between two different datasets (D and $\hat{D}$), including average variation distance and Kullback-Leibler(KL) divergence.

$$\text{Average variation distance} = \frac{1}{|W|} \sum_{r \in W} |M_r(D) - M_r(\hat{D})|$$

$$D_{KL}(D \parallel \hat{D}) = \sum_i D(i) \log\left(\frac{D(i)}{\hat{D}(i)}\right)$$

3.2 Hybrid partitioned data and system models

Hybrid partitioned data. Suppose there $K \cdot S$ parties and a entire dataset D. This dataset employs a K-fold partitioning strategy for horizontal partitioning, where each data slice undergoes an S-channel vertical split. Each party holds only a specific subset of the horizontal data slices and corresponding finite vertical attribute set, adhering to the principle of minimizing data exposure. In brief, the dataset $D = \cup_{i=1}^{K} D_K$ can be viewed as horizontally partitioned among the K data slices, while each data slice $D_i = \cup_{j=1}^{S} D_{i,j}$ can be viewed as vertically partitioned among the S parties. All parties dedicate publish a integrated synthetic data while protecting individual privacy from intrusion. Figure 1 provides a clearer depiction of this example.

ID	Sex	Income	Working status	Degree
	Party 1		Party 2	
1	male	10K	Employment	Bachelor
2	female	5K	Unemployment	Bachelor
3	male	10K	Employment	Bachelor
	Party 3			Party 4
4	male	50K	Employment	Doctor
5	male	10K	Employment	Bachelor
6	female	5K	Unemployment	Bachelor
7	female	5K	Unemployment	Bachelor
8	male	50K	Employment	Master

Figure 1: An example of hybrid partition data.

In the field of private synthetic data generation for multi-party data, it is a common practice to assume that different parties share common identifiers of the users and hold mutually exclusive sets of attributes. If the parties have overlapping attributes, they can send their data schema to the curator to constructs exclusive sets of attributes as a preprocessing step of our solution. Since data schemas are considered public information, such process does not lead to privacy breaches.

System models. Following the common convention in security and privacy research, we adopt a semi-trusted curator (server) in our setting. With the curator's assistance, all parties collaboratively publish an integrated dataset. Specifically, our system model comprises three roles: $K \cdot S$ parties, a semi-trusted curator C, and users. Each party P_i only holds portions of the complete set of data entries and attributes separately. The semi-trusted curator helps the $K \cdot S$ departments release a differentially private version D' of the integrated dataset, which users can then utilize for various data analysis tasks.

In our setting, we adopt the common assumption that both the parties and the curator are semi-trusted (i.e., "honest-but-curious"). This means they will adhere to the protocols correctly but may attempt to infer private information beyond what they are authorized to learn. Our system setting is shown in Figure 2.

3.3 Differential privacy

Differential privacy is a standard paradigm for protecting privacy of individuals. It requires that the change of one entry can only creates a small change of the output distributions.

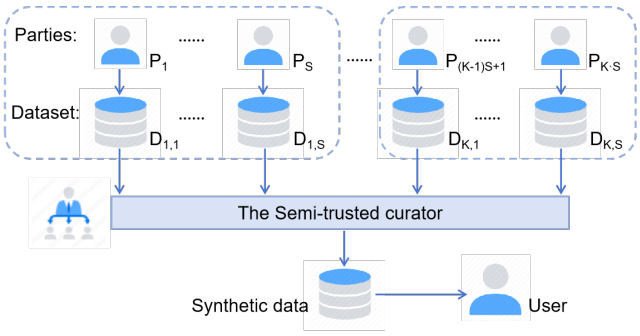


Figure 2: System setting.

Definition 3.3. *(Differential Privacy (DP) [7]). A randomized mechanism $\mathcal{M}: \mathcal{X}^n \rightarrow \mathcal{R}$ satisfies (ϵ, δ)-differential privacy if for any two adjacent inputs D, D' and for any subset of outputs $S \subseteq \mathcal{R}$, it holds that*

$$P_r[\mathcal{M}(D) \in S] \leq e^{\epsilon} P_r[\mathcal{M}(D') \in S] + \delta\,.$$

Our algorithm is defined using Zero Concentrated Differential Privacy (zCDP), an alternative formulation of differential privacy that offers better composition properties when integrating multiple differentially private building blocks. We convert to (ϵ, δ) guarantees when necessary.

Definition 3.4. *(Zero Concentrated Differential Privacy (zCDP) [4]). A randomized mechanism $\mathcal{A}: \mathcal{X}^n \rightarrow \mathcal{R}$ satisfies ρ-zero concentrated differential privacy (ρ-zCDP) if for all neighboring datasets D, D' and for all $\alpha \in (1, \infty)$:*

$$\mathcal{D}_\alpha\left(\mathcal{A}(D) \parallel \mathcal{A}(D')\right) \leq \rho\alpha.$$

where $\mathcal{D}_\alpha\left(\mathcal{A}(D) \parallel \mathcal{A}(D')\right)$ is α-Rényi divergence between the distributions of $\mathcal{A}(D)$ and $\mathcal{A}(D')$.

Lemma 3.1. *(Composition [4]) Two randomized mechanisms $\mathcal{A}_1$ and $\mathcal{A}_2$ satisfy ρ_1-zCDP and ρ_2-zCDP respectively, their sequential composition $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ satisfy $(\rho_1 + \rho_2)$-zCDP.*

Lemma 3.2. *(zCDP implies Differential Privacy[4]) If randomized mechanism $\mathcal{A}$ provides ρ-zCDP, then $\mathcal{A}$ is $\left(\rho + 2\sqrt{\rho \log(1/\delta)}, \delta\right)$-differentially private for any $\delta > 0$.*

There are several important approaches for satisfy differential privacy. The Gaussian mechanism perturbs the numerical output by adding Gaussian noise, and the exponential mechanism is used to select the approximate best among a discrete set of alternatives.

Lemma 3.3. *((zCDP of the Gaussian Mechanism [4]). The Gaussian mechanism which answers query with random noise $\mathcal{N}(0, \Delta^2\sigma^2\mathrm{I})$ satisfies $\left(1/2\sigma^2\right)$-zCDP.*

Here, Δ is the global sensitivity, and $\mathcal{N}\left(0, \Delta^2\sigma^2\mathrm{I}\right)$ denotes a multi-dimensional random variable sampled from the normal distribution with mean 0 and standard deviation $\Delta\sigma$.

Lemma 3.4. *(zCDP of the Exponential Mechanism [5]) The Exponential Mechanism satisfies $\frac{\epsilon^2}{8}$-zCDP.*

Lemma 3.5. *(Parallel combination property [22]) Suppose the data D can be divided into multiple independent and disjoint subsets, $\{D_1, D_2, \ldots, D_n\}$. A set of random algorithms $\{M_1, M_2, \ldots, M_n\}$ working ontheabove-mentioned subsetsrespectively. Wherein, M_i $(1 \leq i \leq n)$ satisfes the ϵ_i-differential privacy on the subset D_i. Then, the set of random algorithms, $\{M_1, M_2, \ldots, M_n\}$, can achieve $\max\{\epsilon_i\}$ differential privacy on the dataset D.*

4 METHOD

In this section, we introduce RandDPGen, a differentially private synthetic data generation algorithm designed for hybrid partitioned data. Building on the select-measure-generate paradigm, RandDPGen provides distributed marginal measurement techniques to answer marginal queries on hybrid partitioned data with minimal error while ensuring privacy protection. Meanwhile, it incorporates a graphical model to efficiently address the challenges of distribution estimation in high-dimensional data scenarios. Additionally, we also implement several effective strategies to boost the utility of the synthetic data in RandDPGen.

4.1 Distributed marginal measurement

In this section, we propose distributed marginal measurement, which enables parties to collaborate with the semi-honest curator to measure a given marginal query on hybrid partitioned dataset. Specifically, we further consider the partitioned data in more detail, treating it as a combination of horizontal and vertical partitioned data. The dataset is first horizontally partitioned into K data slices, and then each data slice is partitioned vertically into S parties. We mainly discussed the approach for privately measuring vertically partitioned data. For horizontally partitioned data, we can simply aggregate the noisy measurements to obtain the results of entire data.

For a given marginal query r, a local dataset $D_{i,j}$ $\{1 \leq i \leq K, 1 \leq j \leq S\}$ faces two distinct scenarios when responding: one where the data attributes it hold perfectly contain the attributes from the marginal query r, meaning their intersection equals the query attributes; and another where the intersection does not fully cover all the attributes of marginal query, resulting in an intersection smaller than the attributes of r.

We divide the marginal query r into S sub-queries r_j $\{1 \leq j \leq S\}$ to address the two scenarios described above, where the attributes of r_j correspond to the intersection of the data attributes held by the current party and original marginal query r. For each sub-query r_j, we will obtain a indicator vector $v_{i,j}$ of the same length as the number of records in a local data. This vector represents the real measurements of sub-query r_j on the local data $D_{i,j}$, where a value = 1 indicates that the query condition is matched, and 0 otherwise. Additionally, due to the presence of a semi-trusted setting, the indicator vectors need to be perturbed using the Gaussian mechanism under differential privacy. This ensures the security of query responses submitted to the semi-honest curator. Finally, a curator can answer the original marginal query r of vertically partitioned data by computing the inner products of S perturbed indicator vectors. The overall process is shown in the Algorithm 1.

For simplicity, we use M^* to represent the marginal measurements of hybrid partitioned data in semi-trusted setting.

Furthermore, we further observed that, under a semi-honest setting, noise of Gaussian mechanism is added to measurements

Algorithm 1 Distributed marginal measurement

Require: hybrid partitioned data $D = \{D_{1,1}, D_{1,2}, \ldots, D_{K,S}\}$; a marginal query r; privacy parameter ρ.

for Horizontally partitioned data $D_i \in D$ **do**
  for Vertically partitioned data $D_{i,j} \in D_i$ **do**
    Capture the intersection $A_{i,j}$ of the attributes between $D_{i,j}$ and r
    if $A_{i,j} \neq \emptyset$ **do**
      Divide sub-query r_j by attributes $A_{i,j}$
      Allocate privacy budget for r_j: $\rho_j = \frac{|A_{i,j}|}{|r|}\rho$
      Calculate indicator vector $v_{i,j} = M_{r_{i,j}}(D_{i,j})$ using sub-query r_j
      Generate noisy indicator vector $\tilde{v}_{i,j}$ by $\mathcal{N}\left(0, 1/2\rho_j\right)$
  end for
  Noisy marginal measurement on vertically partitioned data: $M_r^*(D_i) = \prod_{j \in S} \tilde{v}_{i,j}$
end for
Noisy marginal measurement on entire data: $M_r^*(D) = \sum_{i \in K} M_r^*(D_i)$

of each sub-query. As the number of parties increases, aggregated noise leads to significant error accumulation. We provide an upper bound of noise in distributed marginal measurement:

Theorem 4.1. *Let D, r, ρ be as defined in Algorithm 1, and let $M_r^*(D)$ denote the marginal measurement of hybrid partitioned data D in the semi-trusted model, we have:*

$$M_r^*(D) \leq M_r(D) + \mathcal{N}\left(0, \frac{KS^2}{2\rho}\right).$$

The proof is defered into the Appendix as the page limitation. According to Theorem 4.1, we obtained an upper bound for the error caused by distributed marginal measurement algorithm.

In practical, when the attributes contained in the marginal r are not excessively large, local datasets held by parties may not share any common attributes with it. As a result, the distributed marginal measurement M_r^* can still obtain reasonably performance.

4.2 Graph model for distribution estimation

The select-measure-generate paradigm aims to find a optimal data distribution through an iterative process, culminating in the production of optimal synthetic data. This form can be expressed as follows:

$$\hat{\mathrm{p}} \in \mathrm{argmin}_{\mathrm{p} \in \mathcal{S}} L(\mathrm{p})$$

Here, $\mathcal{S} = \{\mathrm{p} \mid \mathrm{p}(x) \geq 0 \text{ and } \sum_{x \in \Omega} \mathrm{p}(x) = n\}$ is the set of probability distributions over the domain Ω. $L(\mathrm{p})$ is the loss function, which evaluates how well the estimated data distribution p explains the observed measurements. In general, $L(\mathrm{p}) = \|Q(\mathrm{p}) - y\|$, where Q is a query set, y is the noisy measurements of Q on true dataset, and $\|\cdot\|$ is either the ℓ_1 norm or ℓ_2 norm. Minimizing the ℓ_1 norm is equivalent to estimate the data distribution when the noise originates from a Laplace mechanism. Minimizing the ℓ_2 norm is more commonly encountered in the literature, and it corresponds to estimating the distribution for Gaussian mechanism. In this paper, we estimated data distribution p by minimizing ℓ_2 norm. However, existing select-measure-generate algorithms represent the data distribution in vector form, they fail to scale to high dimensions because the size of p grows exponentially with dimensions.

We address this challenge using graphical-model based estimation algorithm called Private-PGM[21], which refines the optimization problem and construct data distributions through graphical-model and maximum entropy principle. It can generate high-quality data distributions even on low-dimensional noise measurements, effectively avoiding the computational burden caused by high dimensionality.

In this paper, we consider the estimation algorithm as a black box, presenting an interface for updating data distributions within our mechanism. Below, we provide a concise summary of this algorithm along with an overview of its three key utilities. Private-PGM takes as input a collection of noisy marginals derived from the sensitive data, in the format of a list of tuples (r_i, y_{r_i}, σ_i) for $i = 1, \ldots, k$, where $y_{r_i} = M_{r_i}(D) + \mathcal{N}(0, \sigma_i^2 \mathbb{I})$.

Distribution Estimation. The core of Private-PGM revolves around an optimization problem, the objective of which is to identify a distribution p that provides the "best explanation" for the noisy observations y_{r_i}.

$$\hat{\mathrm{p}} \in \arg\min_{\mathrm{p} \in \mathcal{S}} \sum_{i=1}^{k} \left\| M_{r_i}(\mathrm{p}) - y_{r_i} \right\|$$

When the observations y_{r_i} are corrupted with independent and identically distributed (i.i.d.) Gaussian noise, the problem of estimating the underlying parameters can be framed as a maximum likelihood estimation (MLE) problem[21]. However, in high-dimensional settings, directly solving this MLE problem through convex optimization over the scaled probability simplex becomes computationally intractable due to the curse of dimensionality. Private-PGM addresses this challenge by leveraging a key insight: the objective function depends on the parameter vector p only through its marginals. This allows Private-PGM to avoid the need to explicitly represent the full high-dimensional distribution. Instead, it exploits the fact that one of the minimizers of the problem can be represented as a graphical model, which provides a compact and efficient representation of the distribution, enabling computationally feasible optimization even in high-dimensional domains.

Synthetic Data Generation. Given an estimated graphical model $\hat{\mathrm{p}}$, Private-PGM introduces a method for generating synthetic tabular data that closely approximates the target distribution. Instead of directly sampling from $\hat{\mathrm{p}}$, which can introduce high variance, Private-PGM employs a randomized rounding procedure. This approach provides a more stable and lower-variance alternative, ensuring that the synthetic data accurately reflects the statistical properties of the original distribution while maintaining computational efficiency.

Junction Tree Size. The time and space complexity of Private-PGM is intricately linked to the measured marginal queries, with the size of the junction tree induced by these queries being the primary factor. Although a detailed understanding of junction tree construction is not required for this paper, it is important to note

that Private-PGM provides a callable function, JT-SIZE, which allows users to evaluate the size of the junction tree in megabytes. The runtime of distribution estimation is approximately proportional to this value. However, if arbitrary marginals are measured, the size of the junction tree can grow exponentially, potentially exceeding memory limits and resulting in impractical runtimes. This underscores the importance of carefully selecting measured marginals to maintain computational feasibility.

4.3 RandDPGen

We integrate the distributed marginal measurement and graphical-model based distribution estimation algorithm into the the select-measure-generate paradigm, and innovatively develop the RandDPGen for generating private synthetic data for hybrid partitioned data. The overall process is shown in the Algorithm 2. Moreover, we meticulously fine-tune various components of RandDPGen to enhance its performance in generating synthetic data, with the specifics outlined below:

Algorithm 2 RandDPGen

Require: hybrid partitioned data $D = \{D_{1,1}, D_{1,2}, \ldots, D_{K,S}\}$; workload W; private parameter ρ; hyperparameter $Maxsize$, e; iterative rounds T; privacy allocation parameter $\alpha, \beta \in (0, 1)$.

1: Initialize distribution p_t using Algorithm 1
2: Let $\rho_0 = \rho/T$, $\rho_{used} = 0$, and $t = 0$
3: Refine workload $W^* = \{r \mid r \in W \text{ and } n_r \leq maxsize\}$
4: Let optimal privacy budget $\rho_{r \in W^*} = \frac{|r|^{2/3}}{\sum_{r \in W} |r|^{2/3}} \cdot \alpha\rho$ and weight $w_{r \in W^*} = \sum_{s \in W} |r \cap s|$ for $r \in W^*$
5: **while** $\rho_{used} < \rho$ **do**
6: $t = t + 1$
7: $\rho_{used} \leftarrow \rho_{used} + \rho_t$
8: **Select** $r_t \in W^*$ using Algorithm 1 with ρ_r and exponential mechanism with function:
9: $$\Pr[r_t = r] \propto \exp\left(\frac{\sqrt{8(1-\alpha-\beta)\rho_t}}{2} w_r \left(\left\|M_r^*(D) - M_r(\mathrm{p}_{t-1})\right\|_1 - \sqrt{\frac{2}{\pi}}\left(\frac{KS^2}{2\beta\rho_t} + \frac{KS^2}{2\rho_r}\right) n_r\right)\right)$$
10: **Measure** marginal on r_t using Algorithm 1 with $\rho = \beta\rho_t$: $y_{r_t} = M_{r_t}^*(D)$
11: **Estimate** data distribution p_t using Algorithm 3:
12: $$\mathrm{p}_t = \arg\min_{\mathrm{p} \in \mathcal{S}} \sum_{i=1}^{t} \left\|M_{r_i}(\mathrm{p}) - y_{r_i}\right\|_2^2$$
13: **if** $\left\|M_{r_t}(\mathrm{p}_t) - M_{r_t}(\mathrm{p}_{t-1})\right\|_1 \leq \sqrt{KS^2/\beta\rho_t\pi}\, n_{r_t}$ **then**
14: Annealing: $\rho_{t+1} \leftarrow 4\rho_t$
15: **else**
16: $\rho_{t+1} \leftarrow \rho_t$
17: **end if**
18: **if** $(\rho - \rho_{used}) \leq 2\rho_{t+1}$ **then**
19: $\rho_{t+1} = \rho - \rho_{used}$
20: **Generate** synthetic data $\hat{D}$ from p_t

Initial distribution. On the eve of generating synthetic data, a common practice is to sample a uniform distribution across the data domain as the Initial distribution, as demonstrated in the research by [21]. However, it poses a significant drawback: the required computational resources increase exponentially with the dimensionality of the private data. We adopted the graph model to solve this problem, which utilizes an unoptimized initial graph model to estimate the marginals in the workload and treat these estimates as our initial distribution (line 1 of the algorithm 2). This approach significantly reduces the requirement of computational resources, achieving effective both time and space conservation. It not only optimizes our initialization process but also demonstrates our heuristic response to the challenge of high-dimensional data.

Workload construction and measurement. In this section, we first introduce the construction of workload, then propose an optimal privacy budget allocation scheme to marginals in the exponential mechanism. As a key component of the select-measure-generate paradigm, a workload is a collection of marginal queries defined according to the requirements for synthetic data. Compared to the works of [6, 28] select marginals based on attribute relevance, this approach is advantageous as it comprehensively considers all marginals and prevents wasting the privacy budget on measuring marginals that are irrelevant to the workload.

In the construction of workload, we drew inspiration from McKenna et al. [20], who observed that marginals involving a larger number of queries often exhibit lower counts. Consequently, directly applying noise addition mechanisms to such marginals can significantly compromise data utility. We address this issue by excluding candidate marginals with a larger number of queries from the workload. In this scenario, measuring marginals with fewer queries achieves a better signal-to-noise ratio while still contributing to the estimation of marginal with a larger number within the workload. Specifically, we set a hyperparameter, $Maxsize$, which will be filtered out if the query number of the candidate marginal is higher than this parameter (Line 3). This ensures that RandDPGen will never select poor marginal queries that lead to a synthetic data with low utility.

Furthermore, obtaining all marginal measurements of the hybrid partitioned data is crucial, as they will be utilized by the exponential mechanism to select a valuable marginal r_t. However, given the large number of marginals in the workload, obtaining accurate and reliable measurements is essential. To this end, we adopt the optimal privacy allocation scheme to calculate the optimal privacy budget for each marginal (line 4 of the algorithm 2), thereby minimizing the overall error. It comprehensively consider the relationship between noisy scale, the size of marginals, and privacy budget based on a optimization idea.

Specifically, we utilize the Gaussian mechanism within centralized differential privacy to ensure the privacy protection of marginal measurements. Our goal is to allocate the privacy budget across all marginal queries in a manner that minimizes the overall noise error. To clarify this, we consider the expected ℓ_1 error of Gaussian noise added to a marginal r. In particular, if the dimension of a marginal (number of attributes) is $|r|$, after adding Gaussian noise with scale σ_r, we expect the ℓ_1 error to be approximately $|r|\sqrt{\frac{2}{\pi}\sigma_r}$. Therefore, with a privacy budget ρ_r, the resulting error

for r is approximately $|r|\sqrt{\frac{1}{\pi\rho_r}}$. By filtering out constant values, we formulate the optimization problem as:

$$\text{minimize} \sum_{r\in W} |r|\sqrt{\frac{1}{\rho_r}}, \text{ subject to} \sum_{r\in W} \rho_r = \alpha\rho$$

We can solve this question by constructing the Lagrangian function: $\mathcal{L} = \sum_r \frac{|r|}{\sqrt{\rho_r}} + \lambda \cdot (\sum_r \rho_r - \alpha\rho)$. By taking partial derivative of $\mathcal{L}$ for each of ρ_r, we have $\rho_r = \left(\frac{2\lambda}{|r|}\right)^{-2/3}$. The value of μ can be solved by equation $\sum_{r\in W} \rho_r = \alpha\rho$. As a result, $\lambda = \frac{1}{2} \cdot \left(\frac{\alpha\rho}{\sum_{r\in W} |r|^{2/3}}\right)^{-3/2}$, and we have

$$\rho_r = \frac{|r|^{2/3}}{\sum_{r\in W} |r|^{2/3}} \cdot \alpha\rho$$

In short, allocating the privacy budget proportional to the 2/3-power of the number of dimension minimizes the overall noise error. Building on this result, we allocate a privacy budget ρ_r to each marginal r within the workload and use Algorithm 1 to compute the marginal measurements of the hybrid partitioned data when using the exponential mechanism.

Exponential mechanism. In line 9 of RandDPGen, we introduced three modifications to the quality score function of the exponential mechanism, aiming to more accurately reflect the utility we expect from measuring the selected marginals. Our new quality score is expressed as follows:

$$S(D, r) = w_{r\in W^*} \left(\left\| M_r^*(D) - M_r(\mathrm{p}_{t-1}) \right\|_1 - \sqrt{\frac{2}{\pi}\left(\frac{KS^2}{2\beta\rho_t} + \frac{KS^2}{2\rho_r}\right)} n_r \right)$$

Three are three differences from the general quality score function $S(D, r) = \|M_r(D) - M_r(\mathrm{p}_{t-1})\|$. First, the expression within the parentheses can be interpreted as the expected improvement in ℓ_1 error achievable by measuring this marginal. Instead of directly measuring a dataset $M_r(D)$, we use $M_r^*(D)$ to denote the marginal measurement on hybrid partitioned data, with a privacy budget of ρ_r. This can be achieved by executing Algorithm 1 for each marginal query $r \in W^*$.

Second, in addition to the ℓ_1 error under the current model, we also considered the expected ℓ_1 error when measuring at the hybrid partitioned data. The latter ℓ_1 error consists of two components: the iterative marginal measurements with privacy parameter $\beta\rho_t$ used to update the data distribution, and the marginal measurements with privacy parameter ρ_r used to select a valuable marginal query. To this end, we modify the quality score function to compensate for the impact of above error on expected improvements when selecting marginals. Additionally, by multiplying by n_r, this function imposes a more significant penalty on larger marginals. Moreover, this modification makes the selection criteria "budget-adaptive", as it recognizes that we can afford to measure larger marginals when σ is smaller, and we should prefer smaller marginals when σ is larger.

Finally, we assign different weights to different marginal to capture their importance in the workload. Specifically, we calculate the weight to its quality score using the formula $w_r = \sum_{s\in W} |r \cap s|$ to better reflect the significance of a marginal query r within the workload. This formula captures the degree to which r overlaps with other marginal queries in the workload W. As a consequence, we use the sensitivity $\triangle t = \max_{r\in W} w_r$ to select a candidate set in the exponential mechanism.

This quality score function demonstrates an intriguing trade-off: the penalty term x discourages marginals with more cells, while the weight w favors marginals with more attributes. However, if the inner expression is negative, a larger weight will amplify the negativity, making the marginal much less likely to be selected.

Annealing. Furthermore, we introduced the annealing algorithm proposed by Ryan et al. in RandDPGen (Line 5 and 13-19), and its utility have been proven in their work[20]. This innovative strategy dynamically adjusts the privacy budget based on the model's actual performance, enhancing the quality of the synthetic data while saving computational time. Additionally, it eliminates the need to manually tune the hyperparameter T, alleviating a commonly overlooked burden for practitioners.

Specifically, the annealing process gradually increases the privacy budget for the next round if the amount of information learned by graph model under the current budget is insufficient. The annealing condition is activated when the difference between $M_{r_t}(\mathrm{p}_t)$ and $M_{r_t}(\mathrm{p}_{t-1})$ is small, indicating that little information was learned in the previous round. Follow the work of [20], we set the annealing threshold as $\sqrt{KS^2/\beta\rho_t\pi}n_{r_t}$. If this condition is satisfied, we let ρ_t be quadrupled, which implies that the noisy scale is halved and privacy budget is doubled in the next round. This method offers two significant advantages. First, when the quality scores of all candidate marginals are generally low, increasing the value of ρ_t can enhance the quality scores of more candidates, allowing for a more precise selection of marginals to measure. Second, in cases where high-quality candidate marginals are heavily affected by noise, increasing the value of it provides effective compensation. We assume that the algorithm will run for T rounds and conservatively initialize ρ_t using this parameter. This serves as an upper bound for the number of rounds it will run, but in practice, the number of rounds will be significantly fewer.

5 THEOREM ANALYSES

5.1 Privacy analyses

We represent the worst-case privacy guarantee of the RandDPGen. Let $\alpha, \beta \in (0, 1)$ be a privacy allocation hyperparameter (higher values of α and β allocate more privacy budget to the initial and iterative noisy measurement):

Theorem 5.1. *The RandDPGen satisfies ρ-zCDP with privacy parameter ρ, and $\left(\rho + 2\sqrt{\rho\log(1/\delta)}, \delta\right)$- Differential privacy for any $\delta > 0$.*

Proof. There are three steps in RandDPGen that depend on the sensitive data: marginal measurement for workload when using the exponential mechanism, privately select a marginal and iterative measurement used to estimate distruibution. The first step satisfies $\alpha\rho$-zCDP, where the total privacy budget $\alpha\rho$ is optimally

allocated to all marginals from the workload, and their noisy measurements are privately submitted to the semi-honest curator. The remainder of RandDPGen runs until the budget is consumed. Each step of this algorithm involves one invocation of the exponential mechanism, and one invocation of the Gaussian mechanism. According to Lemma 3.1 to 3.4, round t of RandDPGen is ρ_t-zCDP for $\rho_t = \beta\rho_t + (1-\alpha-\beta)\rho_t$. Note that at round t, $\rho_{uesd} = \sum_{i=0}^{t} \rho_i$, and by Algorithm 4, it suffices to show that ρ_{used} never exceeds ρ. Specifically, to ensure that the privacy budget used by RandDPGen matches ρ, a special condition (line 8 in Algorithm 4) is used in the annealing algorithm to check whether the remaining budget is insufficient for two rounds with the current parameters. If this condition is satisfied, the algorithm will put all the remaining privacy budget in the final round of execution. After iterations, RandDPGen concludes the loop and then generates private synthetic data.

It is worth noting that, when measuring the hybrid partitioned data, the aggregation of all marginal measurements from data slices with a privacy parameter ρ still satisfies ρ-zCDP, according to the *Lemma*3.5. Meanwhile, in constructing the indicator vector, we allocate the privacy parameter ρ to parties based on the composition theorem to perturb the indicator vector, ensuring that the aggregation of these vector satisfies ρ-zCDP. □

5.2 Error analyses

In this section, we solve the uncertainty quantification of synthetic data generated by RandDPGen, and give guarantees for marginals in the (downward closure of the) workload. We analyze the uncertainty caused from both the noisy measurements with Gaussian noise, and the marginals selected by the exponential mechanism. Follow the work of McKenna et al.[20], we divide the analysis into two cases: the "easy" case, where we can obtain unbiased answers for a particular marginal, and the "hard" case, where we cannot. The details and corresponding theoretical guarantees(Corollary 5.1 and Corollary 5.2) are as follows:

The Easy Case: Selected Marginal Queries. A marginal query r is "selected" whenever $r \subseteq r_t$ for some t. We can obtain an unbiased estimate of $M_r(D)$ by y_t defined in Algorithm 3, and the corresponding variance of that estimate. Further, if multiple t satisfy the condition above, we can reduce the variance through using a weighted average of these estimates.

Theorem 5.2. *(Weighted Average Estimator). Let $r_1, \ldots, r_t$ and $y_1, \ldots, y_t$ be as defined in Algorithm 2, and let $R = \{r_1, \ldots, r_t\}$. For any $r \in R_+$, there is an unbiased estimator $\bar{y}_r = f_r(y_1, \ldots, y_t)$ such that:*

$$\bar{y}_r \sim \mathcal{N}\left(M_r(D), \bar{\sigma}_r^2\right) \text{ where } \bar{\sigma}_r^2 = \sum_{\substack{i=1 \\ r \subseteq r_i}}^{t} \frac{n_{r_i}\sigma_i^2}{n_r}.$$

where, σ_i^2 is the aggregated variance introduced by measuring marginal r_i on hybrid partitioned data.

The proof of Theorem 5.2 is deferred into the Appendix as the page limitation. Meanwhile, this theorem allows us to easily bound its error with, as we show in Theorem 5.3.

Theorem 5.3. *(Confidence Bound). Let $\bar{y}_r$ be the estimator from Theorem 5.2. Then, for any $\lambda \geq 0$, with probability at least $1 - \exp\left(-\lambda^2\right)$, we have:*

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

According to the triangle inequality and the noise upper bound from Theorem 4.1, we have:

Corollary 5.1. *Let $\hat{D}$ be any synthetic dataset, and let $\bar{y}_r$ be the unbiased estimator from Theorem 5.2. Then with probability at least $1 - \exp\left(-\lambda^2\right)$:*

$$\left\|M_r(D) - M_r\left(\hat{D}\right)\right\|_1 \leq \left\|M_r\left(\hat{D}\right) - \bar{y}_r\right\|_1 + \left(\sqrt{\log 2 \cdot n_r} + \lambda\right)\sqrt{\frac{KS^2 t}{\beta\rho} \cdot \max_{i \in t} n_{r_i}}.$$

When we plug in the realized values we get a concrete numerical bound that can be interpreted as a (one-sided) confidence interval. In general, we expect $M_r\left(\hat{D}\right)$ to be close to $\bar{y}_r$, so the error bound for $\hat{D}$ will not be that much larger than that of $\bar{y}_r$.

The Hard Case: Unselected Marginal Queries. In this section, we analyze the error of marginals that are not selected during the execution of RandDPGen. This error can be bounded by leveraging the exponential mechanism to combine the selected marginal queries. Theorem 5.4 quantifies the uncertainty of unselected marginals relative to p_{t-1}.

Theorem 5.4. *Let $\rho_t, \rho_r, r_t, y_t, W^*, \mathrm{p}_t, \alpha, \beta$ be as defined in Algorithm 2, and let $\triangle_t = \max_{r \in C_t} w_r$. For any $r \in C_t$, with probability at least $1 - e^{-\lambda_1^2/2} - e^{-\lambda_2}$, we have*

$$\|M_r(D) - M_r(\mathrm{p}_{t-1})\|_1 \leq w_r^{-1} \cdot B_r + \lambda_1\sqrt{KS^2 n_r/2\rho_r}.$$

where, B_r is equal to:

$$\underbrace{w_{r_t}\left\|M_{r_t}(\mathrm{p}_{t-1}) - y_{r_t}\right\|_1}_{\substack{\text{measurement error} \\ \text{on } r_t}} + \underbrace{\sqrt{\frac{2}{\pi}\left(\frac{KS^2}{2\beta\rho_t} + \frac{KS^2}{2\rho_r}\right)}\left(w_r n_r - w_{r_t} n_{r_t}\right)}_{\substack{\text{relationship between selected} \\ \text{and unselected marginals}}} + \underbrace{\frac{2\triangle_t}{\sqrt{8(1-\alpha-\beta)\rho_t}}\left(\log\left|W^*\right| + \lambda_2\right)}_{\substack{\text{uncertainty from} \\ \text{exponential mechanism}}}.$$

The proof of Theorem 5.4 is defered into the Appendix as the page limitation. In addition, we can get the error B_r from the output of RandDPGen, and then provide a bound to quantify the uncertainty in the form of a one-sided confidence interval. According to the triangle inequality and Theorem 5.4, this inequation can be extended to provide a guarantee for $\hat{D}$.

Corollary 5.2. *Let $\hat{D}$ be any synthetic dataset, and let B_r be as defined in Theorem 5.5. Then with probability at least $1 - e^{-\lambda_1^2/2} - e^{-\lambda_2}$,:*

$$\left\|M_r\left(D\right)-M_r\left(\hat{D}\right)\right\|_1 \leq \left\|M_r\left(\hat{D}\right)-M_r\left(\mathrm{p}_{t-1}\right)\right\|_1 + w_r^{-1}\cdot B_r + \lambda_1\sqrt{KS/N}\sigma_r\sqrt{n_r}.$$

We expect p_{t-1} to be close to $\hat{D}$, especially for larger values of t, making this bound often comparable to the original bound on p_{t-1}.

6 EXPERIMENTS

In this section, we empirically evaluate our RandDPGen approach on Adult and Big-datasets. In particular, we first introduce the dataset and experimental setup. We then show linear and non-linear performance of private synthetic data generated by RandDPGen on hybrid partitioned datasets. Furthermore, we compare the performance of RandDPGen with existing algorithms, demonstrating that our method can also achieve superior performance on both vertically and horizontally partitioned datasets.

6.1 Datasets and setup

Datasets. We adopt Adult[3] and Big5[16] two real datasets in our experiments. Adult dataset obtained from the 1994 US Census includes 15 attributes and about 48, 842 individual tuples. Big5 consists of the answers to fifty survey questions on a five point scale, and includes 19, 719 individual tuples with 57 attributes. Further, we convert each categorical attributes of above datasets into binary attribute set by following the preprocessing steps outlined in [31].

To compare the performance of RandDPGen with existing approaches, we vertically partition the attributes and horizontally partition the records among different parties hybrid to accommodate the distributed multi-party setting. Similar trends under different partitionings are observed.

Tasks. Following the experimental settings outlined in [28], we evaluate the above approaches using both linear and non-linear analytical tasks. We present the linear performance of synthetic data in terms of α-way marginals and KL divergence, as well as its non-linear performance in support vector machine classification. These tasks are widely used to evaluate the quality of synthetic data. Specifically, we first analyze 2-way and 3-way marginals on two real-world datasets and use the average total variation distance to quantify the errors between the all noisy marginals and their noise-free versions. Then, we further demonstrate the utility of the synthetic data using KL divergence, which provides a robust metric to evaluate how well the synthetic data captures the underlying structure and distribution of the real datasets.

For the non-linear task, we train SVM classifiers on the synthetic dataset. During the training process, 80 percent of the tuples from the synthetic data are used for training, while the 20 percent tuples from the real dataset are reserved for testing. Each classifier predicts the label attribute of dataset based on its all other attributes. In particular, we train a classifier on Adult to predict whether a individual earns more than $50K$ annually, and a classifier on Big5 to predict whether a individual is a male. Prediction accuracy is evaluated using the misclassification variance, which measures the difference of misclassification rates on the test set between classifiers trained on synthetic dataset and real dataset. Instead of directly using the misclassification rate, the misclassification variance better reflects the differences between synthetic and real data in nonlinear tasks.

Approaches. In the vertical partitioned setting, we demonstrate the utility of RandDPGen by comparing it with DistDiffGen[24], DPLT[28] and PrivBayes[31] from previous works. DistDiffGen is a two-party differentially private data release algorithm on vertically partitioned data for classification tasks. The second approach is DPLT, which is the state-of-the-art of generating synthetic data on multi-party setting. In addition, we adopt the PrivBayes to further demostrate the performance of our approach. It is worth noting that this algorithm is originally designed for centralized settings. We extend it to distributed setting by learning a K-degree Bayesian network on vertically partitioned data and referred to as PDBN.

In the horizontal setting, we adopt two existing works, DP-SUBN[6] and PrivBayes[31], to demonstrate the utility of our algorithm. DP-SUBN is a state-of-the-art of generating synthetic data on multi-party horizontal setting, which includes DP-SUBN^1, DP-SUBN^2 and DP-SUBN^3. We utilize DP-SUBN^3 to conduct our study, denoted as DP-SUBN. Similar to the vertical setting, the parties independently adopt PrivBayes to construct locally private data, then generate the integrated synthetic data. For simplicity, we denote this method as Ind-PrivBayes.

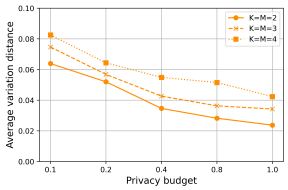
Parameters. Based on the the linear evaluation tasks, we predefined workloads with 2-way and 3-way marginals, respectively, and set the hyperparameter Maxsize set to 10,000 for workload refinement. Subsequently, synthetic data was generated based on both of these workloads. However, after observing a series of experimental studies and careful analysis, we found that synthetic data generated by workload constructed from 2-way marginals performed outstanding performance in certain linear and nonlinear tasks. In light of this discovery, we strategically decided to focus solely on using 2-way workload to construct synthetic data specifically for all subsequent nonlinear tasks. This choice aims to fully leverage the advantages of 2-way marginal workloads in capturing data complexity and maintaining task relevance.

We empirically set the privacy allocation parameters α, β to 0.1 and 0.2, respectively. Additionally, we have defined the hyperparameters $e = K \cdot S$ and $T = 16d$, where d is the number of attribute of D. To make a fair comparsion, we implement these methods across private budgets $\epsilon \in \{0.1, 0.2, 0.4, 0.8, 1.0\}$. All experiments are conducted on an Intel Core i5 with 32 GB RAM. We run each approach 5 times and report the average.

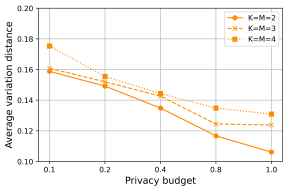
6.2 Performance of RandDPGen on hybrid partitioned data

In this section, we illustrate the utility of RandDPGen on hybrid partitioned data. Specifically, we increase the number of parties by setting K=M=$\{2, 3, 4\}$, and observe their performance in both linear and nonlinear tasks.

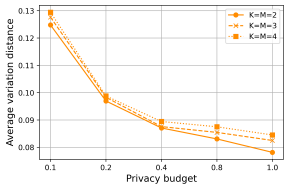
Figure 3 illustrates the average total variation distance of synthetic datasets generated from hybrid partitioned data across various scales. These results reveal that when applied to hybrid partitioned data of different scales, RandDPGen consistently achieves promising outcomes, showcasing its capability to generate high-quality synthetic data from complex and heterogeneous data sources.



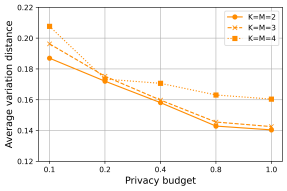
(a) Adult, 2-way marginals



(b) Adult, 3-way marginals



(c) Big5, 2-way marginals



(d) Big5, 3-way marginals

Figure 3: Linear performance of RandDPGen on different number of parties

A key observation from this task is the relationship between data dimensionality and the performance of RandDPGen. Specifically, RandDPGen exhibits a tendency to maintain a relatively low average total variation distance on the high-dimensional Big5 dataset (as shown in Figure 3(c)). This behavior highlights the method's robustness against variations in data dimensionality, indicating that it can adapt effectively to different data dimensionality without compromising the quality of generated synthetic data.

Moreover, the results demonstrate RandDPGen's strong resilience to differences in data scale. Regardless of whether the hybrid data is partitioned at fine or coarse granularity levels, the method maintains stable performance metrics, which further underscores its practical applicability. These findings suggest that RandDPGen not only generates high-quality synthetic data but also offers robustness and scalability across diverse experimental setups.

This analysis shows valuable insights into the capabilities of RandDPGen as a synthetic data generation method. The consistent performance across different scales and dimensions highlights its potential for broader applications in various domains where synthetic data generation is essential. Furthermore, the method's ability to maintain low total variation distances while accommodating complex data structures demonstrates its effectiveness in preserving the inherent characteristics of the underlying data.

Table 1 presents the KL divergence between synthetic and real datasets across different numbers of parties. The relatively low-level observations in these results clearly indicate that our method is highly effective at handling hybrid partitioned data when generating synthetic data. This finding highlights the robust performance of RandDPGen in maintaining data quality while generating synthetic data that closely mirrors real datasets.

By comparing the KL divergence between synthetic and real data on high-dimensional big5 dataset, these results revealed that RandDPGen demonstrates good robustness in handling high-dimensional data, and its synthesized data shows significant effectiveness in managing such data.

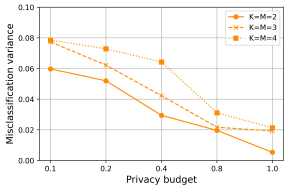
Dataset	Parties	$\epsilon = 0.1$	$\epsilon = 0.2$	$\epsilon = 0.4$	$\epsilon = 0.8$	$\epsilon = 1.0$
Adult	K=S=2	0.0409	0.0249	0.0139	0.0092	0.0072
	K=S=3	0.0567	0.0333	0.0222	0.0165	0.0121
	K=S=4	0.0645	0.0435	0.0289	0.0247	0.0171
Big5	K=S=2	0.0776	0.0572	0.0432	0.0378	0.0306
	K=S=3	0.0810	0.0573	0.0491	0.0384	0.0375
	K=S=4	0.0828	0.0606	0.0504	0.0459	0.0423

Table 1: KL divergence of RandDPGen on different number of parties

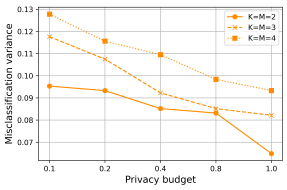
Additionally, from the specific results presented in Table 1, RandDPGen demonstrates stable performance in generating synthetic data as the number of party increases. The low values of KL divergence indicate that the synthesized data shows minimal differences compared to real data, further validating the success of RandDPGen in generating highly similar synthetic data to real-world datasets.

By conducting a comprehensive analysis, this task has demonstrated the unique advantages of RandDPGen in generating high-quality synthetic data. Whether examining overall performance or fine-grained results, the method consistently exhibits strong adaptability and reliability. This is particularly significant for real-world scenarios where synthetic data is essential for training and validation. Additionally, RandDPGen stands out not only for its ability to handle high-dimensional data but also because it maintains low divergence, ensuring that the synthesized data remains as close as possible to real data.

By analyzing Figure 4, which illustrates the misclassification variance of RandDPGen across different scales of parties under varying privacy budgets, we observe a consistent decline in the performance of all synthetic datasets as both K and S increase. However, despite this performance degradation, the overall misclassification variance remains uniformly low, suggesting that RandDPGen exhibits minimal sensitivity to the scaling of the number of parties. These findings further validate the effectiveness of the proposed methods in generating private synthetic data on hybrid partitioned datasets.



(a) Adult, Y = salary



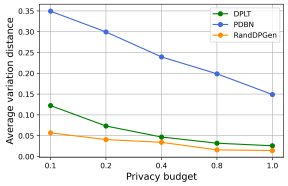
(b) Big5, Y = gender

Figure 4: Non-linear performance of RandDPGen on different number of parties.

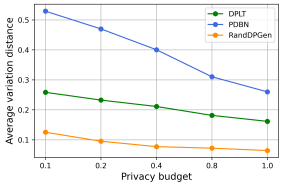
In summary, this experimental study confirms that RandDPGen is a reliable and versatile tool for generating high-quality synthetic data from hybrid partitioned data. Its robustness to different scales and dimensions, coupled with its ability to maintain low evaluation metrics, makes it a valuable asset in scenarios where synthetic data generation is critical.

6.3 Performance of RandDPGen on other partitioned data

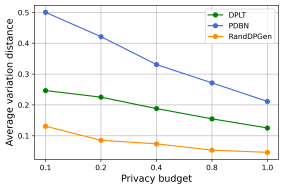
In this section, we evaluate the utility of RandDPGen on vertical and horizontal partitioned datasets by comparing it with existing algorithms on linear and non-linear tasks.



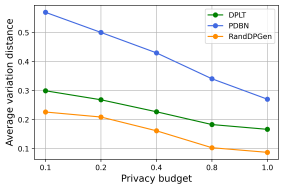
(a) Adult, 2-way marginals



(b) Adult, 3-way marginals



(c) Big5, 2-way marginals



(d) Big5, 3-way marginals

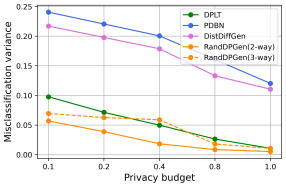
Figure 5: linear performance of RandDPGen, DPLT and PDBN under different privacy budgets.

Vertical partitioned datasets. We first compare RandDPGen with DPLT and PDBN for linear tasks and compare DistDiffGen with DPLT, PDBN and DistDiffGen for only non-linear SVM classification as DistDiffGen is unable to measure α-way marginals. These experiments are conducted in the two-party setting. Figure 5 shows the performance of each algorithm across different privacy budgets on linear analytical tasks. We observed that the synthetic data generated by RandDPGen exhibited a lower average total variation distance compared to DPLT and PDBN, which demonstrates that the select-measure-generate paradigm can leverage a small number of noisy marginals to produce high-quality private synthetic data on vertically partitioned datasets.

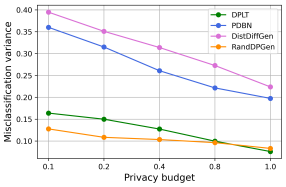
Additionally, we measure the performance of the proposed algorithm on the KL divergence, and the experimental results are presented in Appendix.

Figure 6 exhibits the misclassification variance of the synthetic data generated by each algorithm on Adult and Big5 datasets. It also can be seen that RandDPGen outperforms other algorithms on all the classifiers. In Figure 6(a), we further compare the performance of synthetic datasets generated based on 2-way and 3-way marginals on vertical partitioned data. We can see that the experimental results of RandDPGen based on 3-way marginals are inferior to those based on 2-way marginals, and the misclassification variance of it is higher than DPLT when $\epsilon = 0.4$. Therefore, we decided to focus exclusively on generating synthetic data based on 2-way marginals for all subsequent non-linear tasks.

Additionally, We notice that as the data dimensionality increases (with more attributes), our algorithm was minimally affected, indicating its ability to generate high-quality private synthetic data even for high-dimensional datasets. However, we can observe that RandDPGen delivers promising results in linear tasks (Figure5) but still showed shortcomings in nonlinear tasks (e.g., when $\epsilon = 1.0$, the misclassification variance in Figure 6 is higher than DPLT). This is because our RandDPGen is a workload-aware method that generates synthetic data based on k-way marginals, thus rendering it more sensitive to linear evaluations.



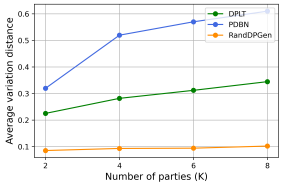
(a) Adult, Y = salary



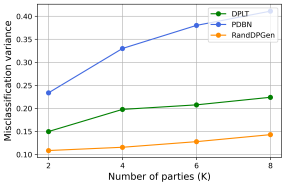
(b) Big5, Y = gender

Figure 6: Non-linear performance of RandDPGen, DPLT, PDBN and DistDiffGen under different privacy budgets.

Furthermore, we experimentally confirm the feasibility of extending RandDPGen to the multi-party setting. Due to DistDiffGen cannot be adapted for the multi-party setting, it is excluded from comparison in this set of experiments. We follow the parameters of [28] and set the privacy budget $\epsilon = 0.2$, similar results are also observed under other privacy budgets. Experimental results of RandDPGen substantially outperforms PDBN and DPLT in both linear and non-linear tasks, since the select-measure-generate paradigm estimates the data distribution of vertically partitioned data with minor noisy marginals and resulting in a synthetic data with superior data utility.



(a) Big5, 2-way marginals

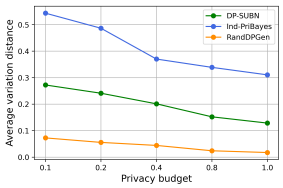


(b) Big5, Y = gender

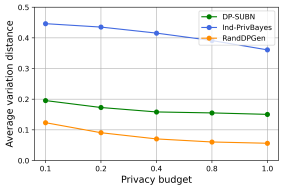
Figure 7: Performance of RandDPGen, DPLT and PDBN under different numbers of parties.

Horizontal partitioned datasets. In this section, we compare RandDPGen with DP-SUBN and Ind-PriBayes for linear and non-linear tasks in the three-party setting. In the linear analysis, Figure 8 shows the average variation distance of each algorithm across different privacy budgets on Adult and big5. We observe that RandDPGen outperforms DP-SUBN and Ind-PriBayes in all marginal measurements, which demonstrates that the select-measure-generate paradigm is capable of producing high-quality private synthetic data on horizontal partitioned datasets by injected noise of minor magnitude.

For non-linear analysis, Figure 9 illustrates the misclassification variance of each approach on the datasets Adult and Big5 under

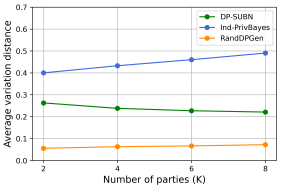


(a) Adult, 2-way marginals

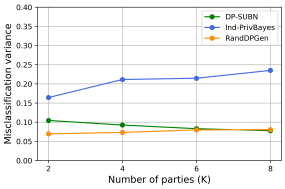


(b) Big5, 2-way marginals

Figure 8: Linear performance of RandDPGen, DP-SUBN and Ind-PriBayes under different privacy budgets.



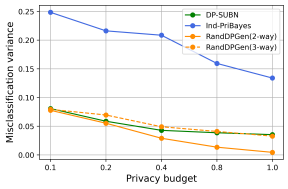
(a) Adult, 2-way marginals



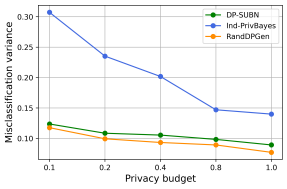
(b) Adult, Y = salary

Figure 10: Performance of RandDPGen, DP-SUBN, and Ind-PriBayes under different numbers of parties.

different privacy budgets. Specifically, Figure 9(a) further demonstrating a similar conclusion that the synthetic data generated by RandDPGen based on 2-way marginals achieve the best data utility on all real datasets.



(a) Adult, Y = salary



(b) Big5, Y = gender

Figure 9: Non-linear performance of RandDPGen, DP-SUBN and Ind-PriBayes under different privacy budgets.

Figure 10 exhibits the utility of varying numbers of parties on on each approach. In particular, Figure 10(a) shows the average variation distance of each approach on 2-way marginals and Figure 10(b) shows the misclassification variance of each approach. We follow the parameters of [31] and set the privacy budget $\epsilon = 0.2$, similar results are also observed under other privacy budgets.

As shown in Figure 10, as parties increases, the performance of RandDPGen and Ind-PriBayes show a decreasing trend, but the degree of RandDPGen is significantly milder compared to Ind-PriBayes. The mild decline of RandDPGen demonstrates that proposed approach effectively controls the amount of injected noise in a multi-party setting. In addition, we also notice that the performance of DP-SUBN improves slightly when K increases on non-linear task, but RandDPGen can still achieve a comparable performance. When the number of parties is not excessively large, RandDPGen is still capable of achieving reasonably good performance.

7 CONCLUSION

In this paper, we study the realistic problem of releasing high-dimensional hybrid partitioned data in the distributed multi-party environment under differentially private protection. To solve this problem, we present RandDPGen, a private synthetic data generation algorithm for high-dimensional hybrid partitioned data. This algorithm achieves marginal measurements on distributed data by proposing the distributed marginal measurements, and alleviate the computational burdens associated with high-dimensional data through introducing the graphical-model based estimation algorithm. Theoretical and experimental analyses demonstrate high data utility of the private synthetic data generated by proposed algorithm.

It is noteworthy that we introduce the key assumption that all parties exhibit homogeneous privacy requirements and allocate same budgets for protecting their local data. While these assumptions simplify the analysis, it is important to note that in real-world scenarios, parties may have varying privacy needs and distinct budget allocations. This limitation motivates our future work on releasing hybrid partitioned data under personalized private protection. Additionally, numerous generative models have demonstrated exceptional performance in the publication of tabular data[17, 19, 29], and the application of these methods to the release of hybrid partitioned data may offer another perspective for our researches.

Furthermore, there are some limitations in our works. As shown in the experiments of varying numbers of parties, as K and S increases, the performance of RandDPGen presents a decreasing trend. Although this trend is mild, it still indicates that the number of parties has a certain impact on RandDPGen. Meanehile, as a workload-aware technique, the data distribution estimated by select-measure-generate paradigm focus on fitting the noisy marginals. However, it does not take into account the diversity of the learned data, which is also a critical aspect of synthetic data.

REFERENCES

[1] Dima Alhadidi, Noman Mohammed, Benjamin CM Fung, and Mourad Debbabi. 2012. Secure distributed framework for achieving ε-differential privacy. In Privacy Enhancing Technologies: 12th International Symposium, PETS 2012, Vigo, Spain, July 11-13, 2012. Proceedings 12. Springer, 120–139.
[2] Noga Alon, Raef Bassily, and Shay Moran. 2019. Limits of private learning with access to public data. Advances in neural information processing systems 32 (2019).
[3] Arthur Asuncion and David Newman. 2007. UCI machine learning repository.
[4] Mark Bun and Thomas Steinke. 2016. Concentrated differential privacy: Simplifications, extensions, and lower bounds. In Theory of Cryptography Conference. Springer, 635–658.
[5] Mark Cesar and Ryan Rogers. 2021. Bounding, concentrating, and truncating: Unifying privacy loss composition for data analytics. In Algorithmic Learning Theory. PMLR, 421–457.
[6] Xiang Cheng, Peng Tang, Sen Su, Rui Chen, Zequn Wu, and Binyuan Zhu. 2019. Multi-party high-dimensional data publishing under differential privacy. IEEE Transactions on Knowledge and Data Engineering 32, 8 (2019), 1557–1571.
[7] Cynthia Dwork. 2006. Differential privacy. In International colloquium on automata, languages, and programming. Springer, 1–12.
[8] Cynthia Dwork and Guy N Rothblum. 2016. Concentrated differential privacy. arXiv preprint arXiv:1603.01887 (2016).
[9] Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, et al. 2020. The pile: An 800gb dataset of diverse text for language modeling. arXiv preprint

arXiv:2101.00027 (2020).
[10] Slawomir Goryczka, Li Xiong, and Benjamin CM Fung. 2013. m-Privacy for Collaborative Data Publishing. IEEE Transactions on knowledge and data engineering 26, 10 (2013), 2520–2533.
[11] Moritz Hardt, Katrina Ligett, and Frank McSherry. 2012. A simple and practical algorithm for differentially private data release. Advances in neural information processing systems 25 (2012).
[12] Yuan Hong, Jaideep Vaidya, Haibing Lu, Panagiotis Karras, and Sanjay Goel. 2014. Collaborative search log sanitization: Toward differential privacy and boosted utility. IEEE Transactions on Dependable and Secure Computing 12, 5 (2014), 504–518.
[13] Wei Jiang and Chris Clifton. 2006. A secure distributed framework for achieving k-anonymity. The VLDB journal 15 (2006), 316–333.
[14] Pawel Jurczyk and Li Xiong. 2009. Distributed anonymization: Achieving privacy for both data subjects and data providers. In IFIP Annual Conference on Data and Applications Security and Privacy. Springer, 191–207.
[15] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. arXiv preprint arXiv:2001.08361 (2020).
[16] Yong-Jun Kim. 2014. [Online]. Available: http://personality-testing.info/$_rawdata/$.
[17] Tennison Liu, Zhaozhi Qian, Jeroen Berrevoets, and Mihaela van der Schaar. 2023. Goggle: Generative modelling for tabular data by learning relational structure. In The Eleventh International Conference on Learning Representations.
[18] Terrance Liu, Giuseppe Vietri, Thomas Steinke, Jonathan Ullman, and Steven Wu. 2021. Leveraging public data for practical private query release. In International Conference on Machine Learning. PMLR, 6968–6977.
[19] Terrance Liu, Giuseppe Vietri, and Steven Z Wu. 2021. Iterative methods for private synthetic data: Unifying framework and new methods. Advances in Neural Information Processing Systems 34 (2021), 690–702.
[20] Ryan McKenna, Brett Mullins, Daniel Sheldon, and Gerome Miklau. 2022. Aim: An adaptive and iterative mechanism for differentially private synthetic data. arXiv preprint arXiv:2201.12677 (2022).
[21] Ryan McKenna, Daniel Sheldon, and Gerome Miklau. 2019. Graphical-model based estimation and inference for differential privacy. In International Conference on Machine Learning. PMLR, 4435–4444.
[22] Frank McSherry and Kunal Talwar. 2007. Mechanism design via differential privacy. In 48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07). IEEE, 94–103.
[23] Ilya Mironov. 2017. Rényi differential privacy. In 2017 IEEE 30th computer security foundations symposium (CSF). IEEE, 263–275.
[24] Noman Mohammed, Dima Alhadidi, Benjamin CM Fung, and Mourad Debbabi. 2013. Secure two-party differentially private data release for vertically partitioned data. IEEE transactions on dependable and secure computing 11, 1 (2013), 59–71.
[25] Noman Mohammed, Benjamin CM Fung, and Mourad Debbabi. 2011. Anonymity meets game theory: secure data integration with malicious participants. The VLDB Journal 20 (2011), 567–588.
[26] Noman Mohammed, Benjamin CM Fung, Patrick CK Hung, and Cheuk-Kwong Lee. 2010. Centralized and distributed anonymization for high-dimensional healthcare data. ACM Transactions on Knowledge Discovery from Data (TKDD) 4, 4 (2010), 1–33.
[27] Jaswinder Singh. 2021. The Rise of Synthetic Data: Enhancing AI and Machine Learning Model Training to Address Data Scarcity and Mitigate Privacy Risks. Journal of Artificial Intelligence Research and Applications 1, 2 (2021), 292–332.
[28] Peng Tang, Xiang Cheng, Sen Su, Rui Chen, and Huaxi Shao. 2019. Differentially private publication of vertically partitioned data. IEEE transactions on dependable and secure computing 18, 2 (2019), 780–795.
[29] Lei Xu, M Skoularidou, A Cuesta-Infante, and K Veeramachaneni. 2019. Modeling tabular data using conditional gan. arXiv 2019. arXiv preprint arXiv:1907.00503 1 (2019).
[30] Mengmeng Yang, Taolin Guo, Tianqing Zhu, Ivan Tjuawinata, Jun Zhao, and Kwok-Yan Lam. 2023. Local differential privacy and its applications: A comprehensive survey. Computer Standards & Interfaces (2023), 103827.
[31] Jun Zhang, Graham Cormode, Cecilia M Procopiuc, Divesh Srivastava, and Xiaokui Xiao. 2017. Privbayes: Private data release via bayesian networks. ACM Transactions on Database Systems (TODS) 42, 4 (2017), 1–41.

A MISSED PROOFS

In this section, we present a comprehensive and detailed account of the proofs supporting our key theorems. These proofs are carefully constructed to demonstrate the logical foundation and validity of each argument, ensuring that all theoretical constructs are rigorously established.

Theorem A.1 (Restatement of Theorem 4.1). *Let D, r, ρ be as defined in Algorithm 1, and let $M_r^*(D)$ denote the marginal measurement of hybrid partitioned data D in the semi-trusted model, we have:*

$$M_r^*(D) \leq M_r(D) + \mathcal{N}\left(0, \frac{KS^2}{2\rho}\right).$$

Proof. We first analyze the noisy measurement of the vertically partitioned data. Let $\frac{|A_{i,j}|}{|r|} = \alpha_j$, we have $\sum_{j=1}^{s} \alpha_j = 1$ and any $\alpha_j \geq 1/s$ $(s \leq S)$.

$$M_r^*(D_i) = \tilde{v}_{i,1} \cdot \cdots \cdot \tilde{v}_{i,S} \leq M_r(D_i) + \sum_{j=1}^{s} \mathcal{N}(0, \frac{1}{2\alpha_j \rho}).$$

Based on the statistical properties of the Gaussian distribution, we have

$$M_r^*(D_i) = M_r(D_i) + \mathcal{N}(0, \frac{1}{2\rho} \sum_{j=1}^{s} \frac{1}{\alpha_j}).$$

Now, we can obtain the upper bound of $M_r^*(D)$ by solving the maximum of function $f(\alpha) = \sqrt{\sum_{j=1}^{s} \frac{1}{\alpha_j^2}}$. We assume that α_j is variable, and $\alpha_1, \ldots, \alpha_{j-1}, \alpha_{j+1}, \ldots, \alpha_s$ are fixed. The partial derivative of $f(\alpha)$ with respect to α_j is negative:

$$\frac{\partial}{\partial \alpha_j}(\sum_{j=1}^{s} \frac{1}{\alpha_j}) = -\frac{1}{\alpha_j^2}.$$

Thus, for any α_j, the value of the function $f(\alpha)$ increases as α_j decreases. Then, we let s take the maximum value S, α_i take the minimum value $1/S$, and obtain $f(\alpha) \leq \sqrt{S^2}$. This equation implies,

$$M_r^*(D_i) \leq M_r(D_i) + \mathcal{N}(0, \frac{S^2}{2\rho}).$$

Next, we analyze the noisy measurement of the horizontally partitioned data. According to the Lemma 3.5, we use the same privacy parameters for horizontal partitioning data, and obtain the upper bound of error from the distributed marginal measurement M_r^*:

$$\begin{aligned} M_r^*(D) = \sum_{i \in K} M_r^*(D_i) &\leq \sum_{i \in K} \left[M_r(D_i) + \mathcal{N}(0, \frac{S^2}{2\rho}) \right] \\ &= M_r(D) + \mathcal{N}(0, \frac{KS^2}{2\rho}). \end{aligned}$$

□

Theorem A.2 (Restatement of Theorem 5.2). *(Weighted Average Estimator). Let $r_1, \ldots, r_t$ and $y_1, \ldots, y_t$ be as defined in Algorithm 2, and let $R = \{r_1, \ldots, r_t\}$. For any $r \in R_+$, there is an unbiased estimator $\bar{y}_r = f_r(y_1, \ldots, y_t)$ such that:*

$$\bar{y}_r \sim \mathcal{N}\left(M_r(D), \bar{\sigma}_r^2\right) \text{ where } \bar{\sigma}_r^2 = \sum_{\substack{i=1 \\ r \subseteq r_i}}^{t} \frac{n_{r_i} \sigma_i^2}{n_r}.$$

where, σ_i^2 is the aggregated variance introduced by measuring marginal r_i on hybrid partitioned data.

Proof. For each $r \subseteq r_i$, we observe $y_i \sim M_r^*(D) = M_r(D) + \mathcal{N}\left(0, \sigma_i^2 \mathbb{I}\right)$. We can use this noisy marginal to obtain an unbiased estimate $M_r(D)$ by marginalizing out attributes in the set $r_i \setminus r$. This requires summing up n_{r_i}/n_r cells, so the variance in each cell becomes $\frac{\sigma_i^2 n_{r_i}}{n_r}$. Moreover, the noise remains normally distributed since the sum of independent normal hybrid variables also follows a normal distribution. We can observe an estimate for each i satisfying $r \subseteq r_i$, and thus we can combine these independent estimates to obtain an unbiased estimator with the stated variance. □

This theorem allows us to easily bound its error with, as we show in Theorem 5.3.

Theorem A.3 (Restatement of Theorem 5.3). *(Confidence Bound). Let $\bar{y}_r$ be the estimator from Theorem 5.2. Then, for any $\lambda \geq 0$, with probability at least $1 - \exp\left(-\lambda^2\right)$, we have:*

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

Proof. Noting that $M_r(D) - \bar{y}_r \sim \mathcal{N}\left(0, \bar{\sigma}_r^2\right)$, the statement is a direct consequence of Lemma 5.1, below. □

Lemma A.1. *[20] Let $x \sim \mathcal{N}\left(0, \sigma^2\right)^n$, then:*

$$\mathbb{E}\left[\|x\|_1\right] = \sqrt{2/\pi} n \sigma,$$

and

$$\Pr\left[\|x\|_1 \geq \sqrt{2\log 2}\sigma n + \lambda\sigma\sqrt{2n}\right] \leq \exp\left(-\lambda^2\right).$$

Theorem A.4 (Restatement of Theorem 5.4). *Let $\rho_t, \rho_r, r_t, y_t, W^*, \mathrm{p}_t, \alpha, \beta$ be as defined in Algorithm 2, and let $\triangle_t = \max_{r \in C_t} w_r$. For any $r \in C_t$, with probability at least $1 - e^{-\lambda_1^2/2} - e^{-\lambda_2}$, we have*

$$\|M_r(D) - M_r(\mathrm{p}_{t-1})\|_1 \leq w_r^{-1} \cdot B_r + \lambda_1 \sqrt{KS^2 n_r / 2\rho_r}.$$

where, B_r is equal to:

$$\underbrace{w_{r_t}\left\|M_{r_t}(\mathrm{p}_{t-1}) - y_{r_t}\right\|_1}_{\substack{\text{measurement error} \\ \text{on } r_t}} + \underbrace{\sqrt{\frac{2}{\pi}\left(\frac{KS^2}{2\beta\rho_t} + \frac{KS^2}{2\rho_r}\right)}\left(w_r n_r - w_{r_t} n_{r_t}\right)}_{\substack{\text{relationship between selected} \\ \text{and unselected marginals}}} + \underbrace{\frac{2\triangle_t}{\sqrt{8(1-\alpha-\beta)\rho_t}}\left(\log\left|W^*\right| + \lambda_2\right)}_{\substack{\text{uncertainty from} \\ \text{exponential mechanism}}}.$$

Proof. According to the exponential mechanism, we have that, with probability at most $e^{-\lambda_2}$, for any $r \in W^*$ we have

$$S(D, r_t) \leq S(D, r) - \frac{2\triangle_t}{\sqrt{8(1-\alpha-\beta)\rho_t}}\left(\log\left|W^*\right| - \lambda_2\right).$$

Let y_r denote the marginal measurement $M_r^*(D)$ on hybrid partitioned data, where the aggregated variance is σ_r^2 and $r \in W^*$. We have

$$S(D, r) = w_r\left(\|y_r - M_r(\mathrm{p}_{t-1})\|_1 - \sqrt{\frac{2}{\pi}\left(\frac{KS^2}{2\beta\rho_t} + \frac{KS^2}{2\rho_r}\right)} n_r\right).$$

Lemma A.2. *[20] Let $a, b \in \mathbb{R}$ and $c = b + z$ where $z \sim \mathcal{N}\left(0, \sigma^2\right)^n$, we have*

$$\Pr[\|a - c\|_1 \leq \|a - b\|_1 - \lambda\sigma\sqrt{n}] \leq \exp\left(-\frac{1}{2}\lambda^2\right).$$

According to Lemma 5.2, with probability at most $e^{-\lambda_1^2/2}$, we have

$$\|y_r - M_r(\mathrm{p}_{t-1})\|_1 \leq \|M_r(D) - M_r(\mathrm{p}_{t-1})\|_1 - \lambda_1 \sigma_r \sqrt{n_r}.$$

Let $E_r = \|M_r(D) - M_r(\mathrm{p}_{t-1})\|_1$, we have

$$S(D, r) \leq w_r\left(E_r - \sqrt{\frac{2}{\pi}\left(\frac{KS^2}{2\beta\rho_t} + \frac{KS^2}{2\rho_r}\right)} n_r - \lambda_1 \sigma_r \sqrt{n_r}\right).$$

Plugging in exponential mechanism and rearranging gives:

$$S(D, r_t) \leq w_r\left(E_r - \sqrt{\frac{2}{\pi}\left(\frac{KS^2}{2\beta\rho_t} + \frac{KS^2}{2\rho_r}\right)} n_r - \lambda_1 \sigma_r \sqrt{n_r}\right) - \frac{2\triangle_t}{\sqrt{8(1-\alpha-\beta)\rho_t}}\left(\log\left|W^*\right| - \lambda_2\right).$$

$$E_r \geq \left(S(D, r_t) + \frac{2\triangle_t\left(\log|W^*| - \lambda_2\right)}{\sqrt{8(1-\alpha-\beta)\rho_t}}\right)/w_r + \sqrt{\frac{2}{\pi}\left(\frac{KS^2}{2\beta\rho_t} + \frac{KS^2}{2\rho_r}\right)} n_r + \lambda_1 \sigma_r \sqrt{n_r}.$$

We denote

$$S(D, r_t) = w_{r_t}\left(\left\|y_{r_t} - M_{r_t}(\mathrm{p}_{t-1})\right\|_1 - \sqrt{\frac{2}{\pi}\left(\frac{KS^2}{2\beta\rho_t} + \frac{KS^2}{2\rho_r}\right)} n_{r_t}\right),$$

and obtain the upper bound of $\sigma_r \leq \sqrt{KS^2/2\rho_r}$ according to Theorem 4.1.

Finally, we combine both two facts via the union bound, along with some algebraic manipulation, yield the stated result. □

B SELECT-MEASURE-GENERATE PARADIGM

As a workload-aware techniques, the select-measure-generate paradigm represents a comprehensive strategy for generating differentially private synthetic data. This approach can naturally be broken up into 3 steps: (1) select a set of queries, (2) measure those queries using the differential privacy noise mechanism, and (3) generate synthetic data that explains the noisy measurements well.

This methodology operates within the predefined workload, denoted as W (alternatively referred to as a set of k-way marginal queries), which encompasses all possible combinations of values across k attributes. Initially, the approach constructs a joint distribution based on the entire attribute domain. Then it runs a greedy, iterative mechanism for workload-aware synthetic data generation.

During each iteration, it privately selects a query q with the highest approximate errors using sophisticated discrete sampling techniques, such as the exponential mechanism. Subsequently, noisy measurements for the selected query q are obtained through differential privacy perturbation mechanisms, including the Gaussian and Laplace mechanisms. These noisy measurements are then utilized, along with effective update functions, to estimate the data distribution. For instance, the PMW (Private Multiplicative Weights) method employs the multiplicative weights mechanism as its update function, while graph models might adopt specific loss functions for this purpose. Ultimately, this approach generate the synthetic data through an estimated data distribution in the last iteration

Algorithm 3 Select-measure-generate Paradigm

Require: Private dataset $X \in \mathcal{X}$; query set Q; privacy parameters ϵ and number of iterations T.

1: Let $\mathbb{A}_0$ be the initial distribution over $\mathcal{X}$
2: Initialize $\epsilon_0 = \frac{\epsilon}{2T}$
3: **for** iteration $i = 1, \ldots, T$: **do**
4: **Select** a query $q_i \in Q$ by sample algorithm such as exponential mechanism
5: **Measure** query q_i with privacy parameters ϵ_0
6: Estimate data distribution **p** by update functions such as multiplicative weights mechanism
7: **end for**
8: **Generate** a private synthetic data by **p**

C DETAIL IN GRAPH-MODEL CONSTRUCTION

In this section, we introduce the construction principle of graphical-model based estimation algorithm in detail[21], which refine the optimization problem and construct data distributions through graphical model (C.1) and maximum entropy principle (C.1). It can generate high-quality data distributions even on low-dimensional noise measurements, effectively avoiding the computational burden caused by high dimensionality.

Definition C.1. *(Graphical model). Let* $\mathrm{p}_\theta(\mathrm{x}) = \frac{1}{Z}\exp\left(\sum_{r\in C}\theta_r(\mathrm{x}_r)\right)$ *be a normalized distribution, where* $\theta_r \in \mathbb{R}^{n_r}$ *and* C *is a collection of marginal measurement. This distribution is a graphical model that factors over the sets* r*, which are the cliques of the graphical model. The vector* $\theta = (\theta_r)_{r\in C}$ *is the parameter vector.*

Theorem C.1. *(Maximum entropy). Given any* $\hat{\mu}$ *in the interior of* $\mathcal{M}$ *there is a parameter vector* $\hat{\theta}$ *such that the graphical model* $\mathrm{p}_{\hat{\theta}}(\mathrm{x})$ *has maximum entropy among all* $\hat{\mathrm{p}}(\mathrm{x})$ *with marginals* $\hat{\mu}$*.*

Specifically, this approach first considers the essence that the loss function of marginal query sets typically depend on the data distribution p only through its marginal measurements $\mu = M_r(\mathrm{p})$, and the marginal vector μ is lower dimensional than p on the marginal r. It then refine the loss function into $L(\mu) = \|\mu - y\|$, and modify the optimization to estimate only the marginals as $\hat{\mu} \in \mathrm{argmin}_{\mu\in\mathcal{M}} L(\mu)$. Here, Q_r is the query constrained on r, and $\mathcal{M} = \{\mu : \exists \mathrm{p} \in \mathcal{S} \text{ s.t. } M_r(\mathrm{p}) = \mu\}$ is the set of all valid marginals. After finding an optimal $\hat{\mu}$, this the algorithm leverages undirected graphical-model and the maximum entropy principle to generate a tractable approximate distribution p_θ. The overall process is shown in the Algorithm 4.

Algorithm 4 Graphical-model based distribution estimation

Require: Run round R; step size η; marginal measurement μ and its noisy measurement y

Let $\theta_0 = 0$ and $\eta_0 = 2$
for $t = 1, \ldots, R$ **do**
  $\mu_t = \mathrm{MARGINAL} - \mathrm{ORACLE}(\theta_{t-1})$
  $\theta_t = \theta_{t-1} - \eta_t \nabla L(\mu_t)$, where $L(\mu_t) = \|\mu_t - y_t\|_2^2$
end for
Generate graphical-model **p** with parameter μ_R and θ_R

Where, MARGINAL-ORACLE refers to a black-box algorithm for computing the marginals μ of a graphical model based on the parameters θ. The variable η_t represents the step size, which is a dynamic constant, and gradually decrease when the loss change is not obvious. Based on this algorithm, we can obtain an approximate data distribution simply by providing the marginal queries and their noise measurements. Within the elect-measure-generate paradigm, the traditional vector-form data distribution can be replaced with a graphical model-based distribution to alleviate the computational burden of high-dimensional data. Furthermore, McKenna et al.[21] also present a accelerated version of Algorithm 3. In practice, we treat this accelerated algorithm as a black box for estimating the approximate distribution in select-measure-generate algorithm, concisely representing it with the following function:

$$\hat{\mathrm{p}} \in \mathrm{argmin}_{\mathrm{p}\in\mathcal{S}} \|M_r(\mathrm{p}) - y\|$$

Here $S = \{\mathrm{p} \mid \mathrm{p}(x) \geq 0 \text{ and } \sum_{x\in\Omega} \mathrm{p}(x) = n\}$ is the set of probability distributions over the domain ω. r and y denote the measured marginal sets and their noise measurements, where the size is k.

D IMPROVED DISTRIBUTED MARGINAL MEASUREMENT

We further observed the accumulated error caused by noisy measurements of each sub-query is increasing with as the number of parties increases. To solve this limitation, we improve the distributed marginal measurement and present a high-utility version with inadequate privacy protection. This improved version effectively alleviates error accumulation caused by the number of parties, ensuring that the noisy marginals measured from the partitioned datasets can achieve consistent performance with those obtained from a entire dataset.

We employ the maximum likelihood estimation to address this challenge (Line 8-9 in Algorithm 5). Specifically, for any given noisy indicator vector $\tilde{v}$, we represent it as $\tilde{v} = v + \mathcal{N}(0, \sigma^2)$, where v is the real response of a sub-query. We will perturb the real results e times to obtain observations $\{\hat{v}^1, \hat{v}^2, \ldots, \hat{v}^e\}$, and maximum likelihood estimate the true vector v. We show the derivation of maximum likelihood estimation below.

For a Gaussian noise with variance σ^2, we have probability density function:

Algorithm 5 Distributed marginal measurement

Require: hybrid partitioned data $D = \{D_{1,1}, D_{1,2}, \ldots, D_{K,S}\}$; a marginal query r; privacy parameter ρ.

for Horizontally partitioned data $D_i \in D$ **do**
  for Vertically partitioned data $D_{i,j} \in D_i$ **do**
    Capture the intersection $A_{i,j}$ of the attributes between $D_{i,j}$ and r
    if $A_{i,j} \neq \varnothing$ **do**
      Divide sub-query r_j by attributes $A_{i,j}$
      Allocate privacy budget for r_j: $\rho_j = \frac{|A_{i,j}|}{|r|}\rho$
      Calculate indicator vector $v_{i,j} = M_{r_{i,j}}(D_{i,j})$ using sub-query r_j
      Repeat e times: generate noisy indicator vector $\tilde{v}^1_{i,j}, \ldots, \tilde{v}^e_{i,j}$ by $\mathcal{N}\left(0, 1/2\rho_j\right)$
      Obtain Maximum likelihood estimation $\hat{v}_{i,j}$ by $\tilde{v}^1_{i,j}, \ldots, \tilde{v}^e_{i,j}$
  end for
  Noisy marginal measurement on vertically partitioned data: $M^*_r(D_i) = \prod_{j \in S} \hat{v}_{i,j}$
end for
Noisy marginal measurement on entire data: $M^*_r(D) = \sum_{i \in K} M^*_r(D_i)$

$$f(\hat{v} \mid v) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(\hat{v}-v)^2}{2\sigma^2}\right)$$

Combine independent observations $\hat{v}^1, \hat{v}^2, \ldots, \hat{v}^e$, the Joint likelihood function:

$$L(v) = \prod_{i=1}^{e} f\left(\hat{v}^i \mid v\right)$$

By taking the logarithm of this function, we obtain:

$$\ell(v) = \log L(v) = \sum_{i=1}^{e} \log f\left(\hat{v}^i \mid v\right)$$

Substituting it into the Gaussian probability density function,

$$\ell(v) = -\frac{e}{2}\log\left(2\pi\sigma^2\right) - \frac{1}{2\sigma^2}\sum_{i=1}^{e}\left(\hat{v}^i - v\right)^2$$

We solve the derivative of $\ell(v)$ and let it equal to zero, then obtain the maximum likelihood estimate of v:

$$v = \frac{1}{e}\sum_{i=1}^{e}\hat{v}^i$$

Based on the statistical properties of Gaussian distribution and Theorem 4.1, we have $M^*_r(D) \leq M_r(D) + \mathcal{N}\left(0, \frac{KS^2}{2\rho e}\right)$. Then, we can set $e = KS^2$ to keep noise at the same level as that of a entire data, and obtain:

$$M^*_r(D) = M_r(D) + \mathcal{N}\left(0, \frac{1}{2\rho}\right)$$

However, we obtain the maximum likelihood estimate of the indicator vector through multiple samplings. While this method undoubtedly enhances data utility, it inevitably violates the principle of differential privacy, which is an inherent characteristic of the differential privacy itself.

Dataset	Algorithm	$\epsilon = 0.1$	$\epsilon = 0.2$	$\epsilon = 0.4$	$\epsilon = 0.8$	$\epsilon = 1.0$
Adult	PDBN	0.4493	0.4198	0.4016	0.3881	0.3865
	DPLT	0.0689	0.0477	0.0392	0.0289	0.0109
	RandDPGen	**0.0285**	**0.0155**	**0.0137**	**0.0042**	**0.0035**
Big5	PDBN	0.6339	0.5963	0.5544	0.4990	0.3813
	DPLT	0.1199	0.0744	0.0485	0.0367	**0.0247**
	RandDPGen	**0.0590**	**0.0465**	**0.0389**	**0.0314**	0.0309

Table 2: KL divergence of RandDPGen, DPLT and PDBN under different privacy budgets.

Dataset	Algorithms	$\epsilon = 0.1$	$\epsilon = 0.2$	$\epsilon = 0.4$	$\epsilon = 0.8$	$\epsilon = 1.0$
Adult	Ind-PriBayes	0.4531	0.4181	0.4106	0.3962	0.3628
	DP-SUBN	0.0672	0.0577	0.0362	0.0255	0.0243
	RandDPGen	**0.0470**	**0.0387**	**0.0178**	**0.0087**	**0.0050**
Big5	Ind-PriBayes	0.7393	0.6207	0.5942	0.5557	0.4717
	DP-SUBN	0.0831	0.0497	0.0482	0.0412	0.0357
	RandDPGen	**0.0535**	**0.0458**	**0.0406**	**0.0343**	**0.0316**

Table 3: KL divergence of RandDPGen, DP-SUBN and Ind-PriBayes under different privacy budgets.

E ABLATION TESTING

E.1 KL divergence of RandDPGen on other partitioned data

Table 2 presents the KL divergence between synthetic and real datasets across different privacy budget on vertical partitioned data. The low-level results of RandDPGen in these results clearly indicate that our method is highly effective at handling hybrid partitioned data when generating synthetic data.

Additionally, by comparing the KL divergence between synthetic and real data on high-dimensional big5 dataset, these results revealed that RandDPGen demonstrates good robustness in handling high-dimensional data, and its synthesized data shows significant effectiveness in managing such data.

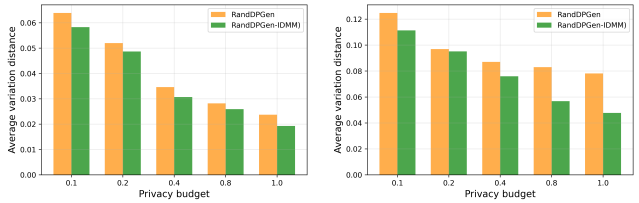
By conducting a comprehensive analysis, this task has demonstrated the unique advantages of RandDPGen in generating high-quality synthetic data. RandDPGen stands out not only for its ability to handle high-dimensional data but also because it maintains low KL divergence, ensuring that the synthesized data remains as close as possible to real data. This is particularly significant for real-world scenarios where synthetic data is essential for training and validation.

According to Table 3, we also obtained a similar result in the study of horizontal partitioned data.

E.2 Ablation testing

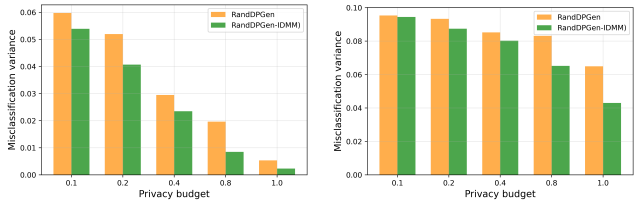
In this paper, we employ an optimized privacy budget allocation scheme(OptPBA) and improved distributed marginal measurement

(IDMM) to reduce the noise scale introduced during running RandDPGen and enhance the quality of synthetic data. Here, we compare the performance of RandDPGen before and after applying these two methods. The number of parties is set to $K = M = 2$ and synthetic datasets generated by 2-way marginals, similar results are also observed under other scales.



(a) Adult, 2-way marginals

(b) Big5, 2-way marginals

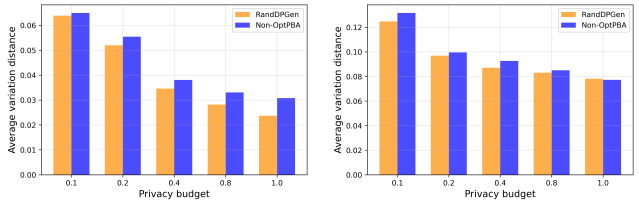


(c) Adult, Y = salary

(d) Big5, Y = gender

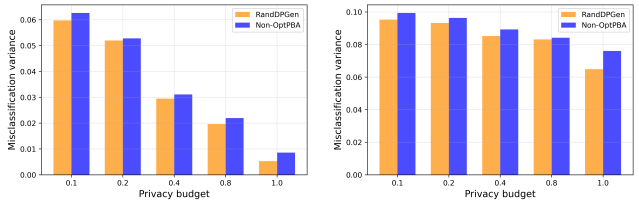
Figure 11: Linear and non-linear performance of RandDPGen before employing IDMM

Figure 11 compares performance of synthetic datasets before and after applying maximum likelihood estimation in improved distributed marginal measurement. We observe that both linear and non-linear performance show a prominent decline when this method are applied, which demonstrates that the proposed method can control noise levels during marginal measurement.



(a) Adult, 2-way marginals

(b) Big5, 2-way marginals



(c) Adult, Y = salary

(d) Big5, Y = gender

Figure 12: Linear and non-linear performance of RandDPGen before employing OptPBA

In addition, Figure 12 compares the performance before and after applying the optimized privacy budget allocation scheme when measuring all marginals of workload. It also can be seen that both linear and non-linear error show a prominent increase when this method are not employed, which prove the effectiveness of this optimized allocation scheme.