ABSyn: An Accurate Differentially Private Data Synthesis Scheme With Adaptive Selection and Batch Processes

Jingyu Jia, Xinhao Li, Tong Li, Zhewei Liu, Chang Tan, Siyi Lv, Liang Guo, Changyu Dong, *Member, IEEE*, and Zheli Liu

***Abstract*— In private data publishing, a promising solution is generating synthetic data that enables any query on the private dataset while satisfying differential privacy. Over the past decade, researchers mainly focused on improving the query accuracy of synthetic data. However, the limitations of existing works restrict them from achieving a better trade-off between accuracy and privacy. In this paper, we propose ABSyn, a novel scheme for differentially private data synthesis. Under the Select-Measure-Generate paradigm, ABSyn has an adaptive mechanism for precisely selecting marginals and follows the batch processes. Our adaptive-batch scheme can provide a well-selected marginal set and the optimal allocation of privacy budget, which makes its synthetic data achieve high accuracy without compromising privacy. We implement an efficient prototype of ABSyn and compare it with existing works by analyzing public datasets. Experimental results show that ABSyn achieves query accuracy on synthetic datasets by a factor of 1.26× and efficiency by a factor of 18.60× over the state-of-the-art scheme on average.**

***Index Terms*— Differential privacy, data synthesis.**

I. Introduction

DIFFERENTIAL privacy (DP) [1] is the gold standard for privacy-preserving data analysis. It plays an important role in measuring privacy leakage in analysis tasks such as aggregates and statistical summaries. Differentially private data synthesis has been widely interested in privacy research [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20], [21], [22], [23], [24]. The synthetic data generated by this technique broadly represents the given data. Given a synthetic dataset, the analyst can perform any query they wish while bounding the privacy leakage of the actual dataset.

Prior research for privacy-preserving data synthesis mainly focuses on providing an accurate synthetic dataset while protecting user privacy. The most appealing differentially private data synthesis schemes [5], [6], [7], [10], [11], [25] follow the *Select-Measure-Generate* paradigm that includes three steps: selecting a set of queries, measuring differential privacy noisy results of these queries, and then generating a synthetic dataset based on the noisy results. This paradigm is renowned for the high query accuracy of its synthetic dataset since schemes following the paradigm capture high-contribution queries with limited interactions with the actual dataset. In terms of data synthesis, the marginals of well-selected queries and the suitable allocation of the privacy budget will introduce little differential privacy noise in the result. Two Select-Measure-Generate schemes [10], [25] won the Differential Privacy Synthetic Data Challenge held by NIST in 2018 [26] and 2020 [27], respectively. Other data synthesis schemes [2], [13], [14], [15], [16], [17], [18] do not follow this paradigm and use deep learning or other methods to construct synthetic datasets. However, these schemes tend to perform poorly in terms of the utility of synthetic datasets.

Prior schemes following the Select-Measure-Generate paradigm have limitations on the select step, which restrict achieving a better trade-off between accuracy and privacy. Some schemes [6], [7], [10] perform the three steps in a *batch*. These schemes take the correlation of low-dimensional marginal, e.g., R-score and its variant, as the selection indicator but overestimate the contribution of selected marginals. Alternatively, the other schemes [5], [11], [25] perform these steps iteratively, such that they can adaptively select a marginal with the most contribution in each round. Unfortunately, the *iteration* mechanism is double-edged in that the adaptive selection cannot guarantee the proper use of the privacy budget and thus results in accuracy loss on synthetic data.

To reduce the accuracy loss above without degrading privacy, we must face a paradox between well-selected marginals and a proper privacy budget allocation. On the one hand, the optimal allocation of the privacy budget can be achieved only if all marginals have been observed. Therefore, the ideal scheme should run the three steps of the Select-Measure-Generate paradigm in a batch sequentially. On the other hand,

Manuscript received 26 December 2023; revised 28 April 2024 and 20 June 2024; accepted 19 August 2024. Date of publication 2 September 2024; date of current version 19 September 2024. This work was supported in part by the Key Program of the National Natural Science Foundation of China under Grant 62032012; in part by the National Natural Science Foundation of China under Grant 62302242, Grant 62272251, Grant 62072132, and Grant 62261160651; and in part by China Postdoctoral Science Foundation under Grant 2023M731802. The associate editor coordinating the review of this article and approving it for publication was Dr. Meng Li. *(Corresponding authors: Tong Li; Siyi Lv.)*

Jingyu Jia, Xinhao Li, Tong Li, Zhewei Liu, Chang Tan, Siyi Lv, and Zheli Liu are with the College of Cyber Science and the College of Computer Science, Key Laboratory of Data and Intelligent System Security, Ministry of Education, Nankai University, Tianjin 300350, China (e-mail: jiajingyu@mail.nankai.edu.cn; asunalxh@mail.nankai.edu.cn; tongli@nankai.edu.cn; liuzhewei@mail.nankai.edu.cn; melonsistan@mail.nankai.edu.cn; lvsiyi@nankai.edu.cn; liuzheli@nankai.edu.cn).

Liang Guo is with Huawei, Beijing 100020, China (e-mail: blue.guo@huawei.com).

Changyu Dong is with the Institute of Artificial Intelligence, Guangzhou University, Guangzhou 510006, China (e-mail: changyu.dong@gzhu.edu.cn).

Digital Object Identifier 10.1109/TIFS.2024.3453175

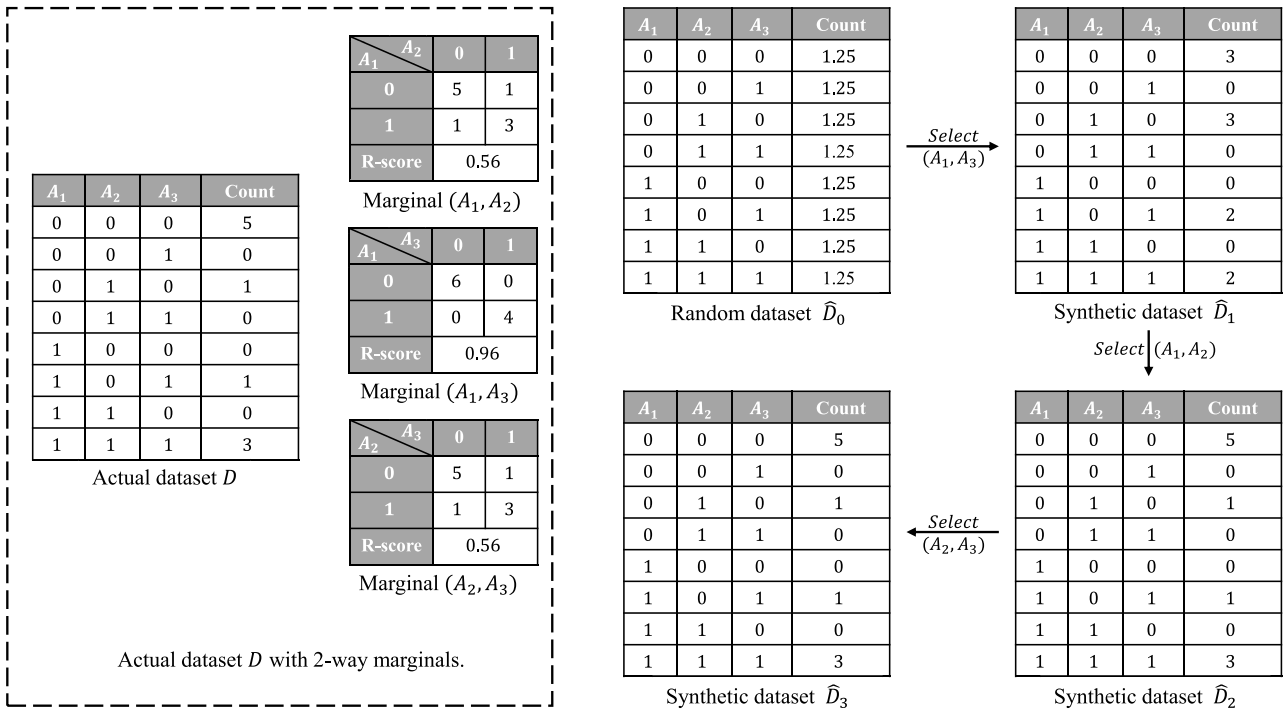


Fig. 1. Example of data synthesis in the non-privacy setting. Given an actual dataset D, sequentially select marginals and update the synthetic dataset in descending order of the correlation indicator R-score.

the best marginals can only be obtained round by round when taking the generated distribution as a reference in an iterative manner. Without the iterative and adaptive observation, it is easy to neglect the influence of marginals on other marginals and thus deviate far from the global optimal results.

In this paper, we propose ABSyn, a novel data synthesis scheme with adaptive marginal selection and batch processes. This adaptive-batch scheme mitigates the dilemma in the select step. Thus, it achieves the high query accuracy of synthetic data and provides a better accuracy-privacy trade-off than state-of-the-art schemes. Specifically, ABSyn benefits from the following two points: First, we propose the Conditional R-score (CR-score), a new indicator to quantify the contribution of marginals in the select step adaptively. Second, we design a new marginal selection algorithm that efficiently computes and utilizes the CR-score to capture the correlation of the actual dataset. The algorithm is compatible with batch processes and can allocate the optimal privacy budget for measuring marginal distributions.

A. ADAPTIVE-Selection Indicator in BATCH Process

Several schemes [6], [7], [25] use R-score to represent the correlation between attributes. This indicator causes an overestimation of the marginals' contribution and introduces additional differentially private noise in synthetic data.

We use a non-privacy example in Figure 1 to show how selected marginals contribute to synthetic data. The actual dataset D has 10 records with three attributes (A_1, A_2, A_3), and each pair of attributes has a high correlation according to the R-score. The synthesis starts with the random dataset $\hat{D}_0$ and sequentially selects marginals in descending order of their correlations. After a marginal is selected, based on the principle of maximum entropy, the dataset is updated to $\hat{D}_1$, $\hat{D}_2$, and $\hat{D}_3$, respectively. Marginals (A_1, A_2) and (A_1, A_3) effectively improve accuracy, while marginal (A_2, A_3) contributes nothing. To make matters worse, if the synthesis were private, selecting marginal (A_2, A_3) would introduce additional noise since more marginals would be measured. This indicator hurts the accuracy of private data synthesis.

The reason for this situation is that, without the whole distribution available for reference, it is difficult to accurately evaluate the contribution of each remaining marginal in the low-dimensional view. Based on this observation, we present the new indicator CR-score as a surrogate function of conditional mutual information to estimate the contribution of marginals. Given a set of selected marginals, we can learn the contribution of each unselected marginal based on this indicator. Therefore, an adaptive selection with the CR-score helps to fully utilize attribute correlation to capture the residual information of marginals.

B. BATCH Synthesis Under ADAPTIVE Selection

Typically, if the synthetic scheme can generate temporary datasets, it can easily calculate the contribution of each marginal under the current state of selected marginals. However, in the batch Select-Measure-Generate paradigm, such a temporary result can not be carried out before finishing the select step. Considering the CR-score is state-dependent, we design an algorithm to approximate this indicator during the adaptive selection. With this algorithm, our scheme can adaptively select high-contribution marginals by solving the optimization problem of minimizing the noise and

CR-score errors. This selection continues several rounds, each of which algorithm selects a marginal to minimize the total error, and the remaining information on marginals is updated. Then, selected marginals are compatible with batch processes, including optimal budget allocation, noise measurement, and synthetic data generation.

We summarize our main contributions as follows.

- **ABSyn**. We propose an adaptive-batch data synthesis scheme, ABSyn, with the advantages of adaptive selection and batch processes and thus provides higher query accuracy than the state-of-the-art schemes.
- **New indicator for adaptive selection in batch synthesis**. We propose a new indicator, CR-score, to accurately evaluate the contribution of marginals. Moreover, we design an algorithm that computes the CR-score for the adaptive selection in the batch data synthesis scheme.
- **Practical implementation of ABSyn**. We implement an ABSyn prototype in which we propose the parallel PGM method that accelerates synthetic data generation. The empirical evaluation of this prototype shows that ABSyn performs better than prior works and improves query accuracy by a factor of at least 1.26× over the state-of-the-art schemes on public datasets. Additionally, ABSyn improves the synthesis efficiency by a factor of 18.60× over the previous most accurate scheme.

II. Related Works

Our paper focuses on the design of an accurate and efficient data synthesis scheme that follows the Select-Measure-Generate paradigm. In this section, we will briefly introduce the data synthesis scheme along this route and some other data synthesis schemes.

Hardt et al. [5] proposed the MWEM algorithm in 2012. The algorithm generates a random dataset and updates the synthetic dataset based on exponential mechanism [28] and multiplicative weight update rule [29]. Zhang et al. [6] propose the R-score to evaluate the correlation in marginals and construct synthetic datasets through Bayesian networks with a limited number of marginals. In 2019, McKenna et al. [9] proposed the PGM to construct synthetic datasets by a probabilistic undirected graphical model. Given a set of noisy marginals, PGM can effectively generate the synthetic dataset. MST [10], PrivMRF [25], and AIM [11] schemes adopt PGM as a component of the generate step to construct synthetic datasets, where the latter two schemes iteratively select marginals and update synthetic datasets, maintaining high utility. In 2021, Zhang et al. [7] proposed the PrivSYN scheme which uses a similar metric to the R-score to describe the correlation between attributes and designs the min-cost flow and the gradually update method to accelerate the update of the dataset distribution.

The above schemes are marginal-based data synthesis schemes and generate synthetic datasets based on the noisy marginals of the actual dataset. Another popular class of privacy-preserving data synthesis schemes is based on deep learning, especially generative adversarial networks [2], [13], [14], [15], [16], [17], [18]. They utilize deep learning to train a deep generative model for generating synthetic datasets and utilize differentially private stochastic gradient descent [30] to preserve privacy during model training. However, the performance of these schemes in empirical testing has shown to be substandard. Compared to marginal-based data synthesis schemes [7], [11], [25], deep learning-based schemes generally fail to demonstrate competitiveness.

III. Preliminary

In this section, we introduce the relevant background on differential privacy and data synthesis to make our work accessible and understandable.

A. Differential Privacy

Differential privacy [31] is a privacy definition for privacy-preserving data analysis. To protect individual privacy, it constrains the impact of a single record on the analysis results. Differential privacy ensures that an individual is not concerned about privacy breaches due to data sharing. This guarantee promotes data sharing since participation in data sharing is mainly irrelevant to privacy leakages.

Definition 1: (Differential Privacy) Let $\varepsilon \geq 0$ *and* $\delta \in [0, 1)$. *A randomized mechanism* $\mathcal{M} : \mathcal{X}^n \to \mathcal{Y}$ *satisfies* $(\varepsilon, \delta)-$*differential privacy if for any* $D, D' \in \mathcal{X}^n$ *that one be obtained by removing a record from the other one, and any* $Y \subseteq \mathcal{Y}$, *it holds that*

$$Pr[\mathcal{M}(D) \in Y] \leq e^{\varepsilon} Pr[\mathcal{M}(D') \in Y] + \delta.$$

1) Global Sensitivity: Sensitivity is an essential notion for differential privacy. The private mechanism achieving differential privacy needs to regulate the noise scale by the sensitivity of the query function.

Definition 2: (Global Sensitivity) Given any function $f : \mathcal{X}^n \to \mathbb{R}^d$, *for any* $X, X' \in \mathcal{X}^n$ *that differs in only one element, the global sensitivity of function* f *is*

$$\Delta_p f = \max_{D, D'} \| f(D) - f(D') \|_p,$$

where $|| \cdot ||_p$ *denotes the* l_p *norm.*

2) Gaussian Mechanism: The Gaussian mechanism [32] achieves differential privacy by adding Gaussian noise to the query results, in which l_2 sensitivity is used to regulate the noise scale. In the scenario of multiple queries on the same dataset, the Gaussian mechanism can provide sufficient privacy protection with a relatively small cost on accuracy.

Theorem 1 (Gaussian Mechanism): Given any function $f : \mathcal{X}^n \to \mathbb{R}^d$, *for any* $D \in \mathcal{X}^n$, *the Gaussian mechanism is defined as*

$$\mathcal{M}_G(D) = f(D) + (\eta_1, \cdots, \eta_d),$$

which satisfies (ε, δ)*-differential privacy, where* η_i *denotes random variables independently drawn from* $N(0, \Delta_2^2 f \sigma^2)$ *with* $\sigma = \frac{\sqrt{2log(1.25/\delta)}}{\varepsilon}$.

3) Composition Theorem of Differential Privacy: The composition theorem of differential privacy allows the computation of cumulative privacy loss for multiple queries to the actual dataset. The theorem guarantees that complex data analysis

schemes combining several differentially private mechanisms are also differentially private.

Theorem 2 (Composition Theorem): Let $\mathcal{M}_1 : D \to R_1$ *be an* $(\varepsilon_1, \delta_1)-$ *differentially private mechanism and* $\mathcal{M}_2 : D \times R_1 \to R_2$ *be an* $(\varepsilon_2, \delta_2)-$ *differentially private mechanism. Then the mechanism* $\mathcal{M} = \mathcal{M}_2(D, \mathcal{M}_1(D))$*is an* $(\varepsilon_1+\varepsilon_2, \delta_1+\delta_2)-$ *differentially private mechanism.*

4) Zero-Concentrated Differential Privacy: As prior works on synthetic data, we focus on zero-Concentrated Differential Privacy (zCDP) [33], an alternate version of differential privacy. zCDP measures privacy leakage of randomized algorithms based on Rényi divergence. Compared to differential privacy, zCDP provides a more accurate privacy analysis for multiple queries on the dataset.

Definition 3: (Zero-Concentrated Differential Privacy) An algorithm $\mathcal{A}$ *satisfies* $\rho-$*zero-concentrated differential privacy if and only if for any two datasets* D *and* D' *that differ by only one record, and for all* $\alpha \in (1, \infty)$ *there is*

$$\mathcal{D}_\alpha(\mathcal{A}(D)||\mathcal{A}(D')) \leq \rho\alpha,$$

where D_α *denotes the Rényi divergence with order* α.

We present the composition theorem for zCDP. Similar to differential privacy, this theorem is used to count the privacy loss of zCDP for multiple queries to the dataset.

Theorem 3 (Composition Theorem for zCDP [33]): Let $\mathcal{M}_1 : D \to R_1$ *be* ρ_1-*zCDP and* $\mathcal{M}_2 : D \times R_1 \to R_2$ *be* ρ_2-*zCDP. Then the mechanism* $\mathcal{M} = \mathcal{M}_2(D, \mathcal{M}_1(D))$*is* $(\rho_1 + \rho_2)-$*zCDP.*

zCDP can be used as a privacy definition to quantify the privacy loss and is more commonly used as a tool to help calculate differential privacy parameters. We present a theorem for the transformation from zCDP to differential privacy.

Theorem 4 (zCDP to Differential Privacy): If a mechanism M *satisfied* $\rho-$*zCDP, it also satisfies* $(\rho + 2\sqrt{\rho \log(1/\delta)}, \delta)-$*differential privacy for any* $\delta > 0$.

Theorems 3 and 4 allow us to analyze a complex differentially private scheme with the help of zCDP. We can count the privacy leakage by zCDP and then convert zCDP to differential privacy.

In this paper, the privacy protection of our scheme comes from the Gaussian mechanism. We present the privacy guarantee of the Gaussian mechanism under zCDP as follows.

Theorem 5 (zCDP of the Gaussian Mechanism): The Gaussian Mechanism that adds noise $N(0, \Delta_2^2 f\sigma^2)$ *to query function* $f(D)$ *satisfies* $\frac{1}{2\sigma^2}-$*zCDP.*

The Rényi divergence of the Gaussian distribution at order α is $\alpha/(2\sigma^2)$, which allows zCDP to provide a tight bound for the Gaussian mechanism. Combination theorems 3 and 4, we can linearly count the privacy leakage of multiple uses of the Gaussian mechanism and transform it into a relatively tight guarantee of differential privacy guarantee.

B. Differentially Private Data Synthesis

1) Dataset: Given a dataset D, private data synthesis aims to generate a synthetic dataset $\hat{D}$ similar to D while satisfying differential privacy. Let the dataset D consist of n records, each of which has d attributes $A = (A_1, \cdots, A_d)$. A differentially private mechanism $\mathcal{M}$ takes D as input and outputs the synthetic dataset $\hat{D}$, which contains $\hat{n}$ records, each of which consists of d attributes as well as in D.

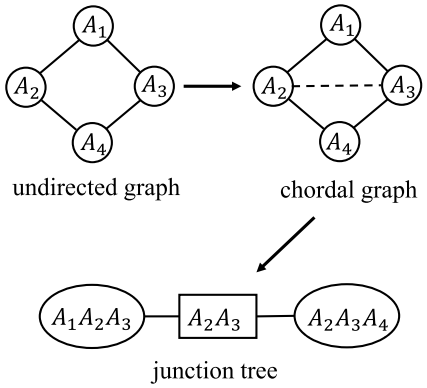


Fig. 2. Example of the PGM method.

2) Marginals: A marginal represents a low-dimensional distribution in the dataset. The state-of-the-art differentially private data synthesis schemes select a set of marginals to capture the distribution of the actual dataset and guide data synthesis. Let $m_i \subseteq A$ be a marginal of the actual dataset, and y_i be the noisy measurement of this marginal by a differentially private mechanism. We use $M = \{m_1, m_2, \cdots\}$ to denote the set of marginals and $Y = \{y_1, y_2, \cdots\}$ to denote the noisy measurements.

Mainstream private data synthesis schemes follow the Select-Measure-Generate paradigm as follows.

- $M \leftarrow Select(D)$. The select step generally selects some low-dimensional marginals from the actual dataset. It takes the dataset D as input and outputs a marginal set M with a high contribution to data synthesis.
- $Y \leftarrow Measure(M)$. The measure step measures the distributions of the selected marginals. It inputs the selected marginal set M and measures the noisy marginal distribution Y via a differentially private mechanism.
- $\hat{D} \leftarrow Generate(Y)$. The generate step generates the synthetic dataset in the same form as the actual dataset. It inputs the noisy measurement Y and outputs the synthetic dataset $\hat{D}$.

C. Private-Probabilistic Graphical Model

Among the existing data synthesis schemes [9], [10], [11], [25], PGM [9] is seen as an effective tool for learning the synthetic dataset distributions. Given a set of noisy marginal measurements of the actual dataset, the PGM uses a probabilistic undirected graphical model to generate synthetic datasets. Assume the dataset has four attributes $\{A_1, A_2, A_3, A_4\}$ and four marginal measurements $(A_1, A_2), (A_1, A_3), (A_2, A_4), (A_3, A_4)$. With each attribute as vertex and noise marginal as edge, PGM constructs the undirected graph as shown in Figure 2.

To improve the inference efficiency, the PGM uses the junction tree for an exact inference of probabilistic undirected graphs. First, the PGM transforms the undirected graph into a chordal graph by adding edge (A_2, A_3). The chordal graph contains two maximal cliques (A_1, A_2, A_3) and (A_2, A_3, A_4).

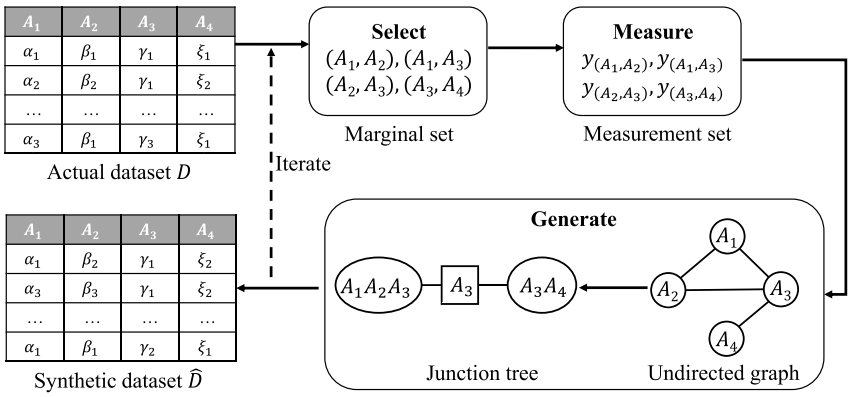


Fig. 3. Data synthesis process.

Then, the junction tree takes maximal cliques (A_1, A_2, A_3) and (A_2, A_3, A_4) as vertices and propagates messages by the intersection (A_2, A_3) to update vertice states.

IV. Prior Works and Their Limitations

In this section, we describe the Select-Measure-Generate paradigm for differentially private data synthesis and the limitations of the state-of-the-art schemes.

A. Select-Measure-Generate Paradigm

The state-of-the-art differentially private data synthesis schemes [5], [6], [7], [10], [25] follow the Select-Measure-Generate paradigm. These schemes generally select and measure some low-dimensional marginals with high correlation by differentially private mechanisms and estimate the distribution of the synthetic dataset with these marginal distributions. Schemes following the Select-Measure-Generate paradigm can effectively utilize the privacy budget to capture correlations between attributes and generate accurate synthetic datasets. To construct synthetic datasets accurately, the design of each step usually follows some intuitions.

1) Select: This step selects some marginals from the actual dataset using a differentially private mechanism. The set of selected marginals is the total information contained in the synthetic dataset and determines the accuracy of the synthetic result. Consider a multi-dimensional dataset with a large domain size. Directly measuring the complete distribution of the dataset is inefficient and introduces a significant amount of noise. To address this problem, previous schemes generally select some low-dimensional marginals to approximate the actual dataset. In this step, the synthesis scheme should select marginals with a high contribution to the data synthesis while reducing the noise introduced during the following steps. If the selection algorithm selects too few marginals, much information in the actual dataset will be lost. Conversely, if it selects many low-contribution marginals, it will introduce significant differentially private noise.

2) Measure: The measure step is used for measuring the selected marginals with a differentially private mechanism. The measurements are constraints for the synthetic datasets. Therefore, its accuracy directly affects the accuracy of the synthetic dataset. A popular approach is to use the Gaussian mechanism to measure the selected marginals. The error of the Gaussian mechanism is relatively low for measuring multiple marginals. In the measure step, schemes need a proper privacy budget allocation strategy to minimize the noise introduced by the Gaussian mechanism.

3) Generate: The generate step outputs a synthetic dataset by learning the noisy measurements of selected marginals. This step is the post-processing step of the differentially private mechanism. The state-of-the-art tool for the generate step is the PGM [9]. Given a set of noisy marginals, the PGM method constructs a probabilistic undirected graph to effectively learn the data distribution and generate a synthetic dataset.

We explain the Select-Measure-Generate paradigm in the context of Figure 3. The actual dataset D has four attributes $A = (A_1, A_2, A_3, A_4)$. First, the scheme selects four marginals $M = \{(A_1, A_2), (A_1, A_3), (A_2, A_3), (A_3, A_4)\}$ from the actual dataset. Then, the scheme uses a differentially private mechanism to measure the noisy measurements $Y = \{y_{(A_1,A_2)}, y_{(A_1,A_3)}, y_{(A_2,A_3)}, y_{(A_3,A_4)}\}$. Finally, the scheme generates a synthetic dataset $\widehat{D}$ via an estimation method (such as the PGM in Figure 3).

B. Batch Method

Batch data synthesis schemes [6], [7], [10] perform the select, measure, and generate steps independently. These schemes select the set of marginals with high correlation in one round. There are two benefits to the batch setup. (1) The scheme has little differential privacy noise in the measure step. Because the scheme can access the final set of selected marginals before the measure step and then properly allocate the privacy budget based on the domain size of marginals. (2) The scheme is efficient because each step needs to be performed only once.

Batch schemes have limitations in quantifying the contribution of marginals. The selection of batch schemes typically uses the correlation of low-dimensional attributes to quantify the contribution of marginals. However, this indicator does not accurately quantify the contribution, leading to the selection of some low-contribution marginals. Selecting low contribution marginals hardly improves the synthetic dataset and introduces additional differentially private noise.

C. Iterative Method With Adaptive Selection

Iterative schemes [5], [11], [25] perform the select, measure, and generate steps iteratively. These schemes construct a

temporary dataset based on selected marginals of the current round and adaptively select a marginal based on the difference between the temporary and actual datasets. These schemes can select the marginal with the highest contribution in each round, thus improving the accuracy of the synthetic dataset.

Iterative schemes have inherent limitations on the allocation of the privacy budget. These schemes cannot acquire the final set of selected marginals in advance. Therefore, they can only heuristically allocate the privacy budget for each round, resulting in accuracy loss in the synthetic dataset. In addition, multiple rounds of iterations increase the computational overhead of these schemes.

V. CR-Score and Adaptive Selection

In this section, we introduce our new indicator, Conditional R-score (CR-score), to evaluate the contribution of marginals in synthetic data. Moreover, we design an algorithm to compute the CR-score during the adaptive selection.

A. Intuition of Marginal Selection

As mentioned in previous sections, the main task of a data synthesis scheme in the select step is to select marginals with high contribution to synthetic data. In the batch scheme, a kind of practical indicator for the selection is the mutual information [6] or a surrogate function of mutual information (R-score) [6], [7], [25], which measures the correlation of low-dimensional marginal. With the help of these indicators, synthesis schemes can select marginals in batch processes.

Intuitively, the R-score-like indicator has a certain rationality and feasibility. Suppose a dataset has two attributes (A_1, A_2). The R-score between A_1 and A_2 is defined as the distance between their joint distribution and the assumed independent distribution:

$$R(A_1, A_2) = \|Pr[A_1, A_2] - Pr[A_1]Pr[A_2]\|_1.$$

R-score has low sensitivity and does not require the construction of a temporary dataset. Based on the R-score, batch schemes [6], [7] can quickly select high-correlation marginals and construct synthetic datasets.

We explain the limitation of R-score-like indicators through an example in the introduction. Here, we further analyze the example to demonstrate the intuition of our indicator. As mentioned above, quantifying the correlation between attributes in isolation does not accurately measure the residual information on marginals. Take Figure 1 as an example. The R-score between the attributes are $I(A_1, A_2) = 0.56, I(A_1, A_3) = 0.96$, and $I(A_2, A_3) = 0.56$, respectively. While the conditional mutual information $I(A_2, A_3|A_1)$ is 0. Blindly selecting all three marginals consumes the privacy budget meaninglessly and increases the difficulty of estimating distribution.

In contrast, iterative schemes can solve this problem with an adaptive method. The scheme constructs a temporary dataset using the set of marginals in the current round to quantify the contribution of the remaining marginals, which naturally achieves the best-contribution selection. Therefore, our accurate synthesis scheme aims to give a new indicator that is conducive to capturing the residual information of marginals and feasible to compute while avoiding the shortcomings of both batch and iterative schemes.

B. CR-Score

In this paper, we propose a new contribution indicator, Conditional R-score (CR-score), that can adaptively measure the contribution of each marginal to synthetic data. Based on the analysis above, there are two requirements for the CR-score. (1) The CR-score should be able to adaptively measure the contribution of the remaining marginals in different sets of selected marginals. (2) The CR-score should be able to be computed without the help of temporary datasets.

We are inspired by the adaptive method of the state-of-the-art iterative scheme, AIM [11], which highlights its advantages in the select step. Consider a dataset with three attributes (A_1, A_2, A_3), and AIM has selected two marginals (A_1, A_2) and (A_1, A_3). Then, AIM uses the PGM method to estimate a temporary dataset for the next select step. Specifically, the PGM method generates temporary datasets $\hat{D}$ based on the principle of maximum entropy. Then, we have

$$\begin{aligned} H_{\hat{D}}(A_1, A_2, A_3) &= H(A_1) + H(A_2|A_1) + H(A_3|A_1, A_2) \\ &\le H(A_1) + H(A_2|A_1) + H(A_3|A_1), \end{aligned}$$

i.e., the entropy of $P(A_1, A_2, A_3)$ is maximum when A_2 and A_3 are conditionally independent given A_1. For the temporary dataset $\hat{D}$, we have

$$\begin{aligned} &I_{\hat{D}}(A_2, A_3|A_1) \\ &= H(A_2|A_1) + H(A_3|A_1) - H_{\hat{D}}(A_2, A_3|A_1) \\ &= H(A_2|A_1) + H(A_3|A_1) - H(A_2|A_1) - H_{\hat{D}}(A_3|A_1, A_2) \\ &= H(A_2|A_1) + H(A_3|A_1) - H(A_2|A_1) - H(A_3|A_1) \\ &= 0. \end{aligned}$$

Informally, if the actual and temporary datasets have a significant difference on a marginal, the conditional mutual information of this marginal should be significant. In other words, the difference of the marginals in the iterative scheme tends to be positively correlated with the conditional mutual information. This indicator describes the residual information of marginals better than the R-score, which tends to be positively correlated with mutual information. The difference between the contribution indicators explains why the iterative selection approach is more effective than batch selection.

We note that a series of mutual information can approximate the conditional mutual information. The attribute set is $A = \{A_1, A_2, A_3\}$ and we set (A_1, A_2) as the candidate marginal. We can establish the following two equations:

$$I(A_1; A_2, A_3) = I(A_1; A_3) + I(A_1; A_2|A_3); \quad (1)$$

$$I(A_1; A_2, A_3) = I(A_1; A_2) + I(A_1; A_3|A_2). \quad (2)$$

From Equation 1 and 2, we have

$$I(A_1; A_2|A_3) = I(A_1; A_2) + I(A_1; A_3|A_2) - I(A_1; A_3). \quad (3)$$

Considering A_1 and A_2 are symmetrical in Equation 3, we can give another expression of $I(A_1; A_2|A_3)$ as follows:

$$I(A_1; A_2|A_3) = I(A_1; A_2) + I(A_2; A_3|A_1) - I(A_2; A_3). \tag{4}$$

Then, we bound $I(A_1; A_2|A_3)$ by an inequality:

$$\begin{aligned} I(A_1; A_2|A_3) &= I(A_1; A_2) - \frac{1}{2}(I(A_1; A_3) + I(A_2; A_3)) \\ &\quad + \frac{1}{2}(I(A_2; A_3|A_1) + I(A_1; A_3|A_2)) \\ &\geq I(A_1; A_2) - \frac{1}{2}(I(A_1; A_3) + I(A_2; A_3)). \end{aligned} \tag{5}$$

Since any mutual information in Equation 5 is positively related to the corresponding R-score, we heuristically indicate $I(A_1, A_2|A_3)$ by a composite R-score of $I(A_1; A_3)$, $I(A_2; A_3)$, and $I(A_1; A_2)$.

Moreover, our CR-score guides the adaptive selection and is compatible with the batch Select-Measure-Generate paradigm. For a marginal set M and an attribute set A, the CR-score of the marginal (A_i, A_j) is denoted as $CR_{A,M}(A_i, A_j)$, where $A_i, A_j \in A$. Here, A is the global attribute set that contains A_i and A_j, while M contains the marginals that affect the correlation of (A_i, A_j). If M contains all possible marginals on A but (A_i, A_j), $CR_{A,M}(A_i, A_j)$ corresponds to the maximum $I(A_1; A_2|A\backslash\{A_1, A_2\})$ on A.

C. Calculation for Adaptive Selection

We carry out the CR-score according to each selection state without referencing temporary synthetic data. For any two attributes A_i, A_j, assume that the attribute domain $d_i \leq d_j$. Then, we have

$$R(A_i, A_j) \leq 1 + \frac{d_i - 2}{d_i}.$$

To ensure that the CR-score is bounded under iterative computation, we normalize the R-score. We propose the normalized R-score as follows:

$$\overline{R}(A_i, A_j) = R(A_i, A_j) \cdot \frac{\min(d_{A_i}, d_{A_j})}{2(\min(d_{A_i}, d_{A_j}) - 1)}, \tag{6}$$

where d_{A_i} and d_{A_j} are the domains of A_i and A_j, respectively. Note that $\overline{R}(A_i, A_j)$ is normalized in $[0, 1]$.

Then, we give a recursive representation of the CR-score of the marginal (A_i, A_j) related to M on A as follows:

$$CR_{A,M}(A_i, A_j) = \overline{R}(A_i, A_j) - \widetilde{R}_{A,M}(A_i, A_j), \tag{7}$$

$$s.t. \widetilde{R}_{A,M}(A_i, A_j) = \max_{(A_i, A_k) \in M} \overline{R}(A_i, A_k) \cdot \widetilde{R}_{A,M}(A_k, A_j), \tag{8}$$

where $\widetilde{R}_{A,M}(A_i, A_j)$ is an intermediate result. The CR-score $CR_{A,M}(A_i, A_j)$ decreases as the normalized R-score of the marginals connecting attributes i and j increases, and we use it to approximate the conditional mutual information.

Algorithm 1 depicts the procedures of computing the CR-score of all marginals. Given an undirected graph G with edge set M, we can compute the CR-score $\{CR_{A,M}(A_i, A_j)\}$ of each attribute pair in G. Each node of G represents an attribute in the actual dataset, and each edge weights the normalized R-score of the corresponding marginal. Our algorithm calculates the maximum weight path from the source node to other nodes. Inspired by the shortest-path algorithm, in each round, the algorithm stores the node with the maximum weight path between the source node, takes the node as an intermediate node, and updates the weights of other nodes. The weight of a node pathway intermediate node to the source node is calculated by multiplication. This setting ensures the algorithm can terminate in a finite number of rounds.

Algorithm 1 CR-Score Computing Algorithm $\mathbf{CR}(G, \overline{R}_A)$

Input: undirected graph G, normalized correlation set $\overline{R}_A = \{\overline{R}(A_i, A_j)\}_{A_i, A_j \in A}$
Output: conditional correlation set $\{CR_{A,M}(A_i, A_j)\}$
Take M as the set of edges in graph G;
For $A_i, A_j \in G$, initialize all $\widetilde{R}(A_i, A_j) \leftarrow 0$;
Let $w_{i,j}$ be the short for $\overline{R}(A_i, A_j)$;
for all nodes A_i in G **do**
    Take A_i as the source;
    Initialize $seen = \{\}$ and $H = \{\}$;
    Push $(A_i, w_{i,i})$ to H;
    while H **do**
        Pop the largest weight tuple $(A_k, w_{i,j})$ from H;
        Update $w_{i,j}$ to $\widetilde{R}_{A,M}(A_i, A_j)$;
        Set $dist_j \leftarrow w_{i,j}$;
        for all node A_k connected to A_j **do**
            $dist_{j,k} \leftarrow dist_j \cdot w_{j,k}$;
            if j not in $seen$ or $dist_{j,k} > seen[k]$ **then**
                $seen[k] \leftarrow dist_{j,k}$;
                push $(A_k, seen[k])$ to H;
            end if
        end for
    end while
end for
for all attribute pair (A_i, A_j) **do**
    $CR_{A,M}(A_i, A_j) \leftarrow \overline{R}(A_i, A_j) - \widetilde{R}_{A,M}(A_i, A_j)$;
end for
Return: $\{CR_{A,M}(A_i, A_j)\}$;

VI. ABSyn Scheme

In this section, we formally describe our ABSyn scheme and provide privacy proof of ABSyn.

The overview of ABSyn is shown in Algorithm 2. Given an actual dataset D, ABSyn translates the differential privacy to zCDP and allocates the privacy budget to the select and measure steps. Then, ABSyn uses the Gaussian mechanism to measure all 1-way marginals of the actual dataset. Next, ABSyn selects the two-way marginals with our marginal selection algorithm. Then, ABSyn allocates the privacy budget, measures the 2-way marginals by the Gaussian mechanism, and performs noise minimization consistency for 1-way and 2-way marginals. Finally, ABSyn efficiently generates the synthetic dataset using the PGM method.

Algorithm 2 Procedure of ABSyn

Input: dataset D, privacy parameters (ε, δ)
Output: synthetic dataset $\hat{D}$
// *Pre-processing*
Converting (ε, δ) to ρ by Theorem 4;
Allocate privacy budget $\{\rho_1, \rho_2, \rho_s\} \leftarrow \rho$;
Measure 1-way marginals Y_1 using $\mathcal{M}_G$ with ρ_1;
//*Select Step*
Select 2-way marginal $M_s \leftarrow$ **Select**(D, ρ_s, ρ_2);
//*Measure Step*
Allocation ρ_2 for each marginal in M_s;
Measure 2-way noisy marginals Y_2;
Consistency processing for noisy marginals Y_1 and Y_2;
//*Generate Step*
Generate synthetic dataset $\hat{D} \leftarrow$ **PGM**(Y_1, Y_2);
Return: synthetic dataset $\hat{D}$;

A. Scheme Details

1) Select Step: Using the CR-score as the reference, we design the selection algorithm to minimize the missing information and noise errors and select the marginals with high contributions to data synthesis. Specifically, the algorithm selecting a marginal will introduce a differentially private noise error. In comparison, the algorithm deselecting a marginal will introduce a missing information error. We compute the missing information error with the help of the CR-score.

For total $d(d-1)/2$ marginals, b_i indicates whether the i-th marginal is selected. The noise error η_i is added to the i-th marginal whose CR-score is $CR_{A,M}(m_i)$. The optimization problem is shown as follows:

$$\begin{aligned} &minimize \sum_{i=1}^{|M|} [\eta_i b_i + CR_{A,M}(m_i)(1-b_i)] \\ &subject\ to\ b_i \in \{0,1\}, \end{aligned} \tag{9}$$

where the target is to minimize the noise error of selected marginals along with the CR-score of absent marginals.

The pseudo-code of the marginal selection algorithm for ABSyn is shown in Algorithm 3. This algorithm is used for solving the optimization problem (Equation 9). In lines 3-8, the algorithm initializes the selected marginal set and measures the R-score by the Gaussian mechanism. In line 9, the algorithm calculates the initial error of unselected marginal. The algorithm selects marginals in lines 11-37 to reduce the total error. In each round, the algorithm constructs an undirected graph based on selected marginals for each unselected marginal m_i (lines 13-16). Next, the selection algorithm calculates the total error of the undirected graph (lines 17-26). In lines 21-22, the algorithm calculates the correlation error for the unselected marginals based on the weighted undirected graph. In lines 23-24, the algorithm calculates the noise error by considering the privacy budget allocation strategy according to the measure step. When the total error cannot be reduced by selecting new marginals, the algorithm breaks and returns the selected marginals (lines 31-33).

In particular, the algorithm triangulates the undirected graph to match the accelerated generation algorithm in Section VII (line 17). This setting ensures that the intersections of maximal cliques of the PGM are selected marginals. For efficiency, we set a threshold τ for maximal cliques of the PGM in this algorithm. The algorithm discards the marginal if the maximal clique size exceeds the threshold.

Algorithm 3 Marginal Selection Algorithm **Select**(D, ρ_s, ρ_m)

1: **Input:** dataset D, privacy parameters ρ_s, ρ_m and clique threshold τ
2: **Output:** selected marginal sets M_s
3: Collect marginal set M and marginal size set c of D;
4: $M_s \leftarrow \emptyset$;
5: $\sigma \leftarrow \sqrt{\frac{d(d-1)}{4\rho_s}}$, $t = 0$;
6: **for all** 2 way marginal $m_i \in M$ **do**
7: $R(m_i) \leftarrow R(m_i) + N(0, \Delta_2^2 f \sigma^2)$;
8: **end for**
9: $E_0 \leftarrow |D| \sum_{m_i} R(m_i)$;
10: Normalize R to $\overline{R}$ by Equation 6;
11: **while** True **do**
12: **for all** unselected marginal m_i **do**
13: Initialize undirected graph G_{t_i};
14: **for all** marginal m_j in $M_s \cup \{m_i\}$ **do**
15: Add edge m_j with $\overline{R}(m_j)$ to G_{t_i};
16: **end for**
17: Triangulated undirected graph G_i to $G_{\Delta t_i}$;
18: Use all edges in $G_{\Delta t_i}$ as selected marginal set M_{t_i};
19: Collect marginal size set c_{t_i};
20: **if** all clique size in $G_{\Delta t_i} < \tau$ **then**
21: $cr_{t_i} \leftarrow$ **CR**$(G_{\Delta t_i}, \overline{R})$;
22: Compute $E_{cor}(t_i)$ with cr_{t_i};
23: Allocate budget ρ_m to ρ_{t_i} using Equation 11;
24: $E_\eta(t_i) \leftarrow \sum_{m_j \in M_{t_i}} \mathbf{c}_{t_{(i,j)}} \sqrt{\frac{1}{\pi \rho_{t_{(i,j)}}}}$;
25: $E_{t_i} \leftarrow E_{cor}(t_i) + E_\eta(t_i)$;
26: **else**
27: Continue;
28: **end if**
29: **end for**
30: $E_t \leftarrow \min E_{t_i}$, $M_t \leftarrow M_{t_i}$;
31: **if** $E_t \geq E_{t-1}$ **then**
32: Break;
33: **else**
34: $M_s \leftarrow M_t$;
35: **end if**
36: $t \leftarrow t + 1$;
37: **end while**
38: **Return:** M_s;

2) Measure Step: In the measure step, we improve the accuracy of the noisy measurements by a privacy budget allocation algorithm and a consistency algorithm for noise variance minimization.

a) Budget allocation algorithm: As mentioned above, ABSyn is a batch scheme that can naturally adopt an optimal privacy budget allocation strategy [7] based on the domain size of selected marginals.

This strategy assigns the privacy budget to minimize the mean absolute error of the Gaussian mechanism. Given marginal set $M = \{m_1, \ldots, m_k\}$ and their domain size $C = \{c_1, \ldots, c_k\}$, the algorithm aims to find a privacy budget allocation by solving the following optimization problem:

$$\begin{aligned} &minimize \sum_i c_i \sqrt{\frac{1}{\pi \rho_i}} \\ &subject\ to \sum_i \rho_i \leq \rho, \end{aligned} \tag{10}$$

where ρ denotes the privacy budget of the measure step, and ρ_i is the privacy budget allocated for marginal m_i.

Equation 10 has a solution when each marginal m_i is assigned a privacy budget as follows:

$$\rho_i = \rho \frac{c_i^{2/3}}{\sum_j c_j^{2/3}}. \tag{11}$$

The proof of Equation 11 can be found in the work of Zhang et al. [7]. By solving the above optimization problem, ABSyn adds different noise scales for each marginal to minimize the overall error in the measure step.

b) Noise minimization consistency process: Given the measurements of selected marginals, ABSyn performs a noise minimization algorithm to make the 1-way and 2-way marginals consistent. Suppose the Gaussian mechanism collects the distribution of $\mathbf{p}_{A_1}$, $\mathbf{p}_{A_1A_2}$ and $\mathbf{p}_{A_1A_3}$. The consistency process projects these distributions onto A_1. Let $|c_{A_1}|$ denote the domain size of A_1, and σ_A denote the standard deviation of A_1. We have

$$var_{A_1} = \frac{1}{\frac{1}{\sigma^2_{A_1}|c_{A_1}|} + \frac{1}{\sigma^2_{A_1A_2}|c_{A_1A_2}|} + \frac{1}{\sigma^2_{A_1A_3}|c_{A_1A_3}|}}.$$

Then we calculate the noise variance minimizing distribution $\overline{\mathbf{p}}_{A_1}$ by the following equation:

$$\overline{\mathbf{p}}_{A_1} = var_{A_1}\left(\frac{\mathbf{p}_{A_1}}{\sigma^2_{A_1}|c_{A_1}|} + \frac{\mathbf{p}_{A_1A_2}}{\sigma^2_{A_1A_2}|c_{A_1A_2}|} + \frac{\mathbf{p}_{A_1A_3}}{\sigma^2_{A_1A_3}|c_{A_1A_3}|}\right).$$

Next, the distribution of $\mathbf{p}_{A_1A_2}$ and $\mathbf{p}_{A_1A_3}$ is updated based on $\overline{\mathbf{p}}_{A_1}$. Because consistency processing is a post-processing of the Gaussian mechanism, we can reduce the error of measurements without costing the privacy budget.

3) Generate Step: To preserve high accuracy on synthetic data, we adopt the state-of-the-art method, PGM, as the generation algorithm. The PGM algorithm constructs the probabilistic undirected graph with attributes as vertices and selected marginals as edges and then trains the graph with marginal distributions as constraints. We use the PGM method to learn the data distribution and construct synthetic datasets effectively.

B. Privacy Analysis

In this section, we give a comprehensive privacy analysis for our scheme. There are three places to access the actual dataset in ABSyn. Therefore, we count the cumulative privacy loss of performing the Gaussian mechanism in the following three places:

1) In line 6 of Algorithm 2, ABSyn measures all 1-way marginals in the pre-processing step.
2) In lines 6-8 of Algirithm 3, our marginal selection algorithm calculates the R-score for all two-way marginals.
3) In line 11 of Algorithm 2, ABSyn measures selected 2-way marginals in the measure step.

First, we convert differential privacy parameters (ε, δ) to zCDP privacy parameter ρ by Theorem 4 and use zCDP to count privacy losses. For the three places with access to the actual dataset, we assign privacy budgets of ρ_1 and ρ_2 to measure 1-way and 2-way marginals and ρ_s to the marginal selection algorithm. Based on the composition theorem for zCDP (Theorem 3), we establish

$$\rho = \rho_1 + \rho_2 + \rho_s.$$

We allocate $\rho_1 = 0.1\rho$ for 1-way marginal measurements, $\rho_s = 0.1\rho$ for the R-score computation, and $\rho_2 = 0.8\rho$ for 2-way marginal measurements.

For 1-way marginal measurements, the privacy budget ρ_1 is evenly distributed among all marginals. If there are d marginals, we assign a privacy budget of ρ_1/d to each marginal. The sensitivity of 1-way marginal measurement is 1. We introduce noise drawn from $N\left(0, \frac{d}{2\rho_1}\right)$ into each bin of each marginal.

For the R-score calculation of the marginal selection algorithm, the privacy budget ρ_s is evenly distributed among the R-score of all 2-way marginals. The sensitivity of R-score is 4. For a total of $\frac{d(d-1))}{2}$ 2-way marginals, we introduce noise drawn from $N\left(0, \frac{4d(d-1)}{\rho_s}\right)$ into each R-score of 2-way marginal.

For 2-way marginal measurements, we assign the privacy budget ρ_2 based on the domains of the selected marginals (Equation 11). If there are k selected 2-way marginals and each marginal m_i is assigned a privacy budget of $\rho_{(2,i)}$, then we introduce noise drawn from $N\left(0, \frac{1}{2\rho_{(2,i)}}\right)$ into each bin of each marginal m_i.

VII. Efficient Implementation

When deploying a data synthesis scheme, we usually encounter a situation: Data synthetic scheme selects a large set of selected marginals at low privacy levels leading to the probabilistic undirected graph of the PGM method becoming dense. If the probabilistic undirected graph contains multiple cliques with large domain sizes, the PGM method takes a long time to train a synthetic dataset. To address this problem in implementing our ABSyn, we propose the parallel PGM method to accelerate model training without compromising accuracy. Our implementation is publicly available on GitHub.[1]

For the generate step, the PGM method uses the junction tree to exact inference of probabilistic graphical models. The junction tree transmits the distribution of each vertice through belief propagation. Since the state-of-the-art scheme provides a

[1] https://github.com/con-fide/ABSyn.git

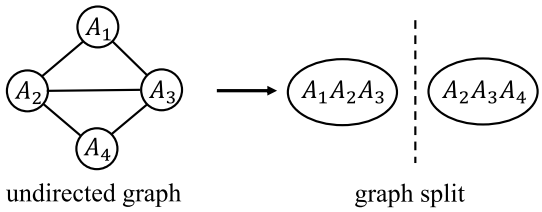


Fig. 4. Graph partition.

flexible marginal selection method, the select step will select more marginals at a low privacy level, resulting in a larger domain size for each maximal clique in the graphical model. In this case, the linear process of belief propagation is not conducive to fast model training.

In addition, belief propagation of the junction tree is not always effective in data synthesis. Suppose the propagation message has already been captured in the measure step, belief propagation will hardly improve the accuracy of this message. Take Figure 4 as an example. The dataset has four attributes $\{A_1, A_2, A_3, A_4\}$ and five high correlation marginals (A_1, A_2), (A_1, A_3), (A_2, A_3), (A_2, A_4) and (A_3, A_4). In each training round, the vertices in junction tree (A_1, A_2, A_3) and (A_2, A_3, A_4) update their distributions based on the noisy measurements of marginals and then send messages to each other through the intersection (A_2, A_3) for belief propagation. Each vertice computes its distribution with the measurements as constraints before belief propagation. Generally, two distributions over (A_2, A_3) are close to the noisy distribution. Thus, even without belief propagation, distributions of the two vertices projected onto (A_2, A_3) are almost identical.

We note that the belief propagation of the junction tree may not improve the query accuracy on synthetic data. Our method is to select intersection marginals between maximal cliques in the select step. Then, we omit each belief propagation process and train each maximal clique in parallel.

Taking Figure 4 as an example. Instead of computing the junction tree, we only need to compute the approximate maximum entropy distributions of nodes (A_1, A_2, A_3) and nodes (A_2, A_3, A_4) individually. Then, we combine all the sub-graphs at the final stage and compute the synthetic dataset distribution.

Algorithm 4 Parallel Private PGM Algorithm **P-PGM**(Y)

1: **Input:** noisy marginal set Y;
2: **Output:** synthetic dataset $\hat{D}$;
3: Generate undirected graph G by marginal Y;
4: **for all** maximal clique in G **do**
5: Train each clique in parallel;
6: **end for**
7: Combine all maximal cliques to generate the synthetic dataset $\hat{D}$;
8: **Return:** $\hat{D}$;

The parallel PGM method for the generate step is shown in Algorithm 4. The algorithm constructs a probabilistic undirected graph based on the selected marginals and then trains each maximal clique of the probabilistic undirected graph in parallel. In an actual deployment, if the number of processes is less than that of maximal cliques, we assign cliques to a limited number of processes based on the domain size of cliques with a greedy algorithm.

VIII. Experiments

In this section, we perform a comprehensive empirical evaluation of the implementation of ABSyn and other representative schemes under multiple public datasets.

A. Settings

In our experiments, all evaluations are performed on a server with CPU Intel(R) Xeon(R) Silver 4208@2.10Ghz, 16 cores, 64 GB of RAM, and 1 TB NVME hard disk.

1) Datasets: Our evaluation includes five public datasets with different features. The characteristics of these datasets are summarized in Table I.

2) Schemes and Parameters: We compare ABSyn with five schemes, MWEM+PGM [9], MST [10], PrivMRF [25], AIM [11], PrivSyn [7]+PGM and DP-GAN [14]. Note that, for a fair comparison, we reproduce the select and measure step of PrivSyn and use PGM, the state-of-the-art generation algorithm, instead of its generate step. We set the privacy parameter as $\varepsilon \in [0.01, 5]$ and $\delta = 10^{-9}$. We repeat each experiment five times and record the average results for each scheme.

B. Evaluation

1) Accuracy on k-Way Marginals: We use the Total Variation Distance (TVD) between the actual dataset and the synthetic dataset as a metric for evaluating the utility of the synthetic dataset across 2-way, 3-way, and 4-way marginals.

Figure 5 illustrates the utility of k-way marginals for each scheme under different conditions. In most instances, ABSyn demonstrates superior accuracy compared to other methods, indicating its ability to effectively preserve the marginal distribution of the actual dataset. On average case, ABSyn outperforms PrivSyn +PGM by a factor of 1.26×, AIM by a factor of 1.26×, PrivMRF by a factor of 1.32x, MWEM+PGM by a factor of 3.74×, MST by a factor of 4.02× and DP-GAN by a factor of 9.08. In extreme cases, ABSyn outperforms PrivSyn+PGM by a factor of 2.01×, AIM by a factor of 1.86×, PrivMRF by a factor of 3.48×, MWEM+PGM by a factor of 84.14×, MST by a factor of 96.44×, and DP-GAN by a factor of 207.56×.

We gather the average ranking for accuracy and the frequencies of best accuracy for each scheme in Tables II and III. Table II describes the average ranking of each scheme in terms of marginal query and classification tasks, which reflects the stability of a scheme across different datasets and different computational tasks. Table III describes the frequency of each scheme exhibiting the best performance across various conditions, indicating how well each scheme performs with specific datasets and privacy settings. According to the findings, ABSyn demonstrates significant superiority in marginal accuracy. Our scheme achieves an average ranking of no more than 1.4 for 2-way, 3-way, and 4-way marginal query

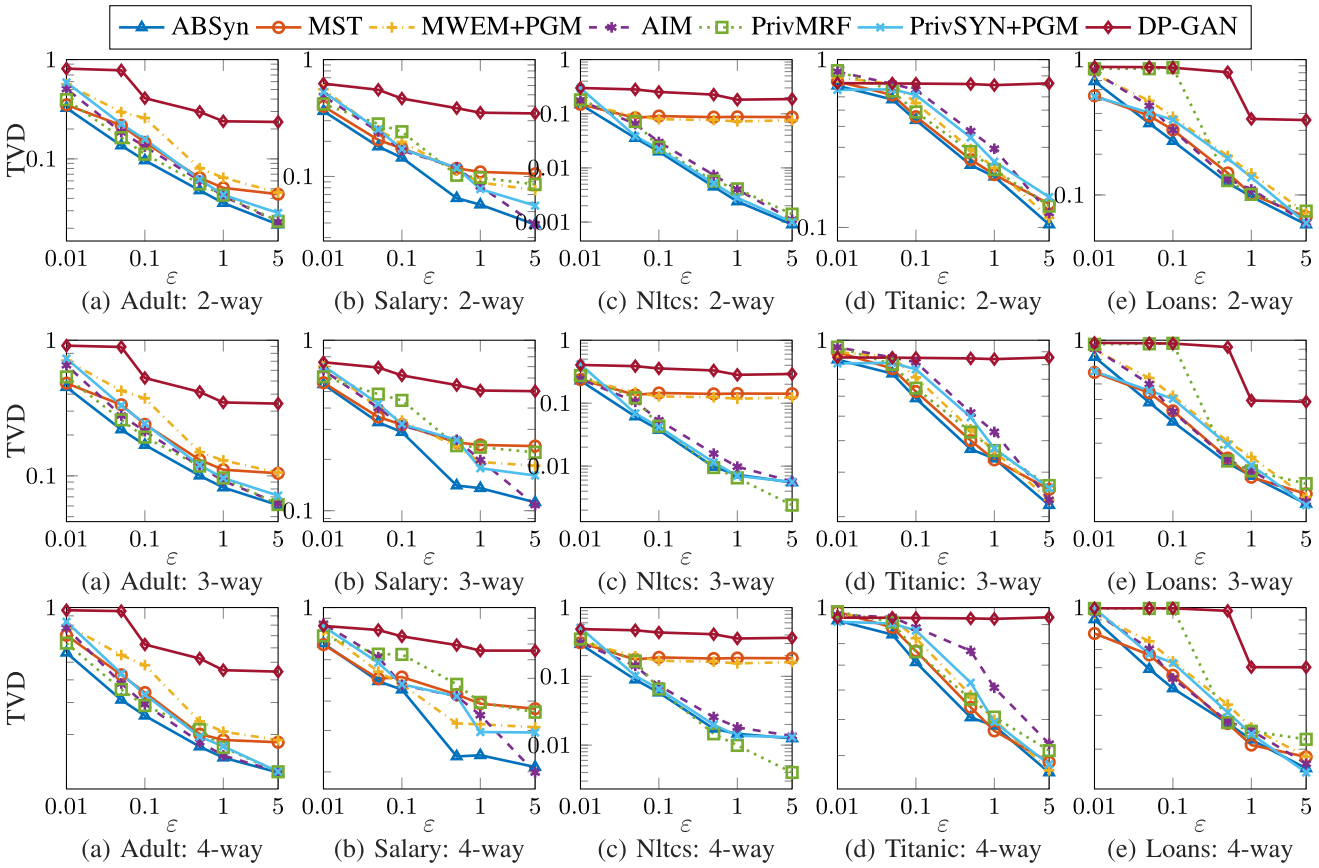


Fig. 5. Marginal error of each scheme.

TABLE I
CHARACTERISTICS OF THE EXPERIMENTAL DATASETS

Dataset	num	attributes	max size	total size
Adult [34]	48842	14	100	6×10^{17}
Salary [35]	135727	9	501	4×10^{15}
Nltcs [36]	21574	16	2	7×10^{4}
Titanic [37]	1304	9	100	3×10^{8}
Loans [38]	42535	50	1001	1×10^{84}

TABLE II
AVERAGE RANKING OF EACH SCHEME

Mechanism	2-way	3-way	4-way	Classification
ABSyn	1.10	1.37	1.40	2.30
AIM [11]	3.7	3.77	3.93	3.33
PrivSyn+PGM [7]	3.77	3.83	3.57	4.37
MST [10]	3.83	3.70	3.70	3.23
PrivMRF [25]	3.90	3.83	4.00	4.27
MWEM+PGM [5]	4.87	4.77	4.60	3.53
DP-GAN [14]	6.83	6.73	6.73	6.97

TABLE III
OPTIMAL FREQUENCY OF EACH SCHEME

Mechanism	2-way	3-way	4-way	Classification
ABSyn	28	21	19	12
AIM [11]	0	1	1	3
PrivSyn+PGM [7]	2	2	2	3
MST [10]	0	3	5	5
PrivMRF [25]	0	3	3	1
MWEM+PGM [5]	0	0	0	6
DP-GAN [14]	0	0	0	0

tasks across all datasets, and shows the highest accuracy in the majority of conditions. The results show that our scheme performs well under various datasets and parameter settings.

AIM, PrivMRF, and PrivSyn employ adaptive marginal selection strategies to increase the number of selected marginals to improve synthetic datasets when the privacy budget is sufficient, especially for the Nltcs dataset, which has a small attribute field. PrivMRF performs well on the accuracy of 3-way and 4-way marginals at large privacy budgets. In contrast, MST and MWEM+PGM adhere to a fixed number of marginals across all conditions. The limited marginal selection approach minimizes noise bias in data synthesis under small privacy budgets or instances of numerous attributes and low sample sizes. However, it also results in a significant loss of information about the actual data under high privacy budget conditions, thereby reducing accuracy. Compared to other schemes, DP-GAN performs significantly worse than the other schemes in most cases.

2) Accuracy on Classification: We also evaluate the utility of the data synthesis schemes on classification tasks. Our approach involves training a classifier using deep learning with the synthetic dataset as the training samples and the actual

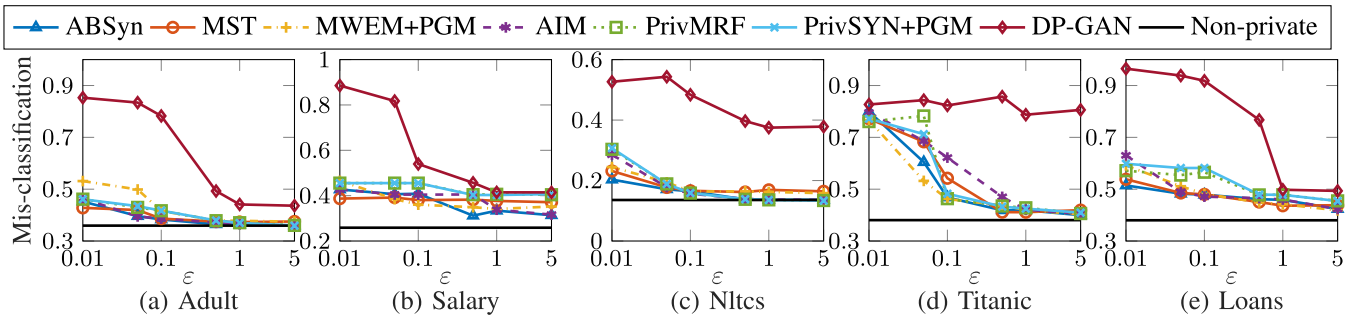


Fig. 6. Mis-classification rate of each scheme.

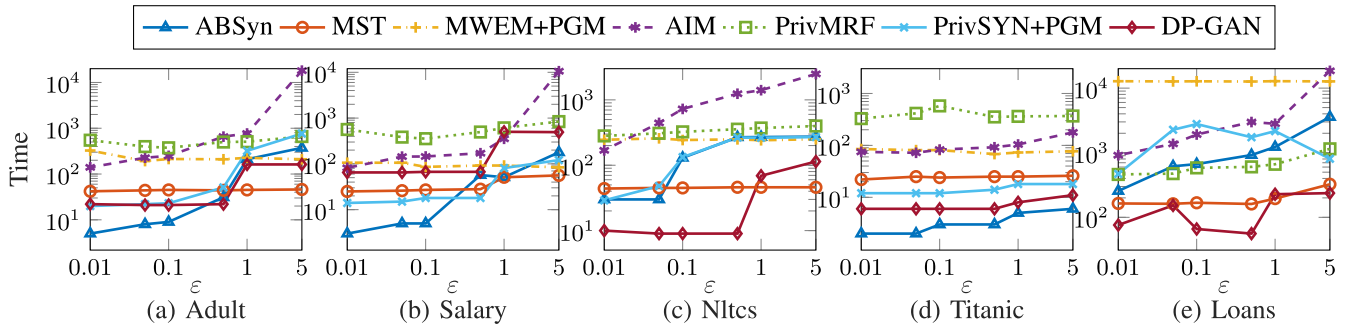


Fig. 7. Running time of each scheme.

dataset as the testing samples. The metric we employed is the mis-classification rate. In addition, we additionally train the classifier with the actual dataset as a non-private scheme.

We implement a multi-class classification model using TensorFlow and Keras, training and evaluating separately for each attribute. The model comprises a fully connected layer with 64 neurons and a ReLU activation function, followed by a 0.1 dropout layer to mitigate overfitting, and a dynamically sized output layer with softmax activation to accommodate the number of classes per attribute, as determined by the range of values for the column. During training, the model is trained over each attribute with a batch size of 64 and 10 epochs.

The average mis-classification rate for each scheme is shown in Figure 6. ABSyn's superiority in the classification task is slightly diminished when compared to the marginal query task, yet it continues to outperform other schemes. On average case, ABSyn outperforms PrivSyn +PGM by a factor of 1.12×, AIM by a factor of 1.05×, PrivMRF by a factor of 1.09x, MWEM+PGM by a factor of 1.06×, MST by a factor of 1.05× and DP-GAN by a factor of 1.87. In extreme cases, ABSyn outperforms PrivSyn+PGM by a factor of of 2.00×, AIM by a factor of 1.41×, PrivMRF by a factor of 1.49×, MWEM+PGM by a factor of 1.30×, MST by a factor of 1.24×, and DP-GAN by a factor of 3.18×.

3) Efficiency: Figure 7 illustrates the runtime performance of each scheme under different conditions. The number of selected marginals varies across ABSyn, PrivSyn+PGM, AIM, and PrivMRF depending on the privacy budget. In conditions with high privacy budgets, these approaches tend to select more marginals, leading to increased training time. In contrast, ABSyn demonstrates reduced runtime efficiency due to batch processes and parallel PGM implementation. It is important to mention that our method takes a relatively large amount of time for the Loans dataset. This is due to the Loans dataset having more attributes, leading to an increased computational cost in the select step.

Both MWEM+PGM and MST select a fixed number of marginals regardless of the privacy budget, resulting in consistent runtimes. Notably, MST utilizes batch processing, while MWEM+PGM follows iterative processing involving multiple times of model training, causing MST to consistently outperform the MWEM+PGM scheme in terms of runtime. DP-GAN has a relatively low runtime and grows as the privacy budget increases.

On average case, ABSyn outperforms PrivMRF by a factor of 47.58×, PrivSyn+PGM by a factor of 2.51×, AIM by a factor of 18.60×, MWEM+PGM by a factor of 16.59×, MST by a factor of 2.93×, and DP-GAN by a factor of 2.96×. In extreme cases, ABSyn outperforms PrivMRF by a factor of 207.00×, PrivSyn+PGM by a factor of 6.00×, and AIM by a factor of 59.00×.

C. Compare to Non-Parallel Version

To validate the effectiveness of our parallel PGM method, we evaluated the non-parallel version of the ABSyn scheme, which uses the original PGM method to generate synthetic data. The performance of ABSyn and non-parallel version in terms of accuracy, classification task, and time overhead is shown in Figures 8, 9 and 10.

Experimental results show that our parallel PGM approach has little effect on the utility of synthetic datasets. The average difference in marginal accuracy and classification between the two schemes does not exceed 0.01. For time overhead, our parallel PGM method shows significant advantages on Adult, Salary, and Titanic datasets. For the Nltcs dataset, which has a field of 2 for each attribute. Both the parallel and non-parallel

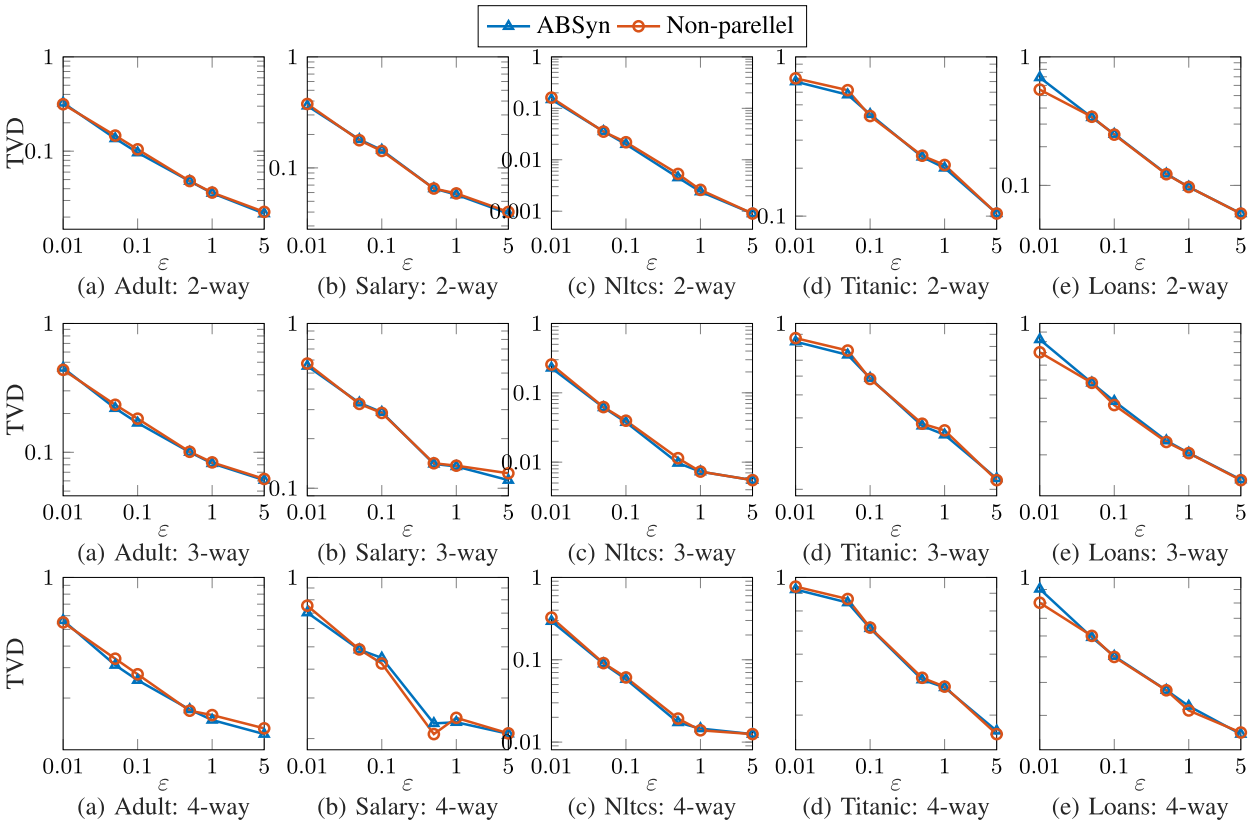


Fig. 8. Marginal error of ABSyn and non-parallel scheme.

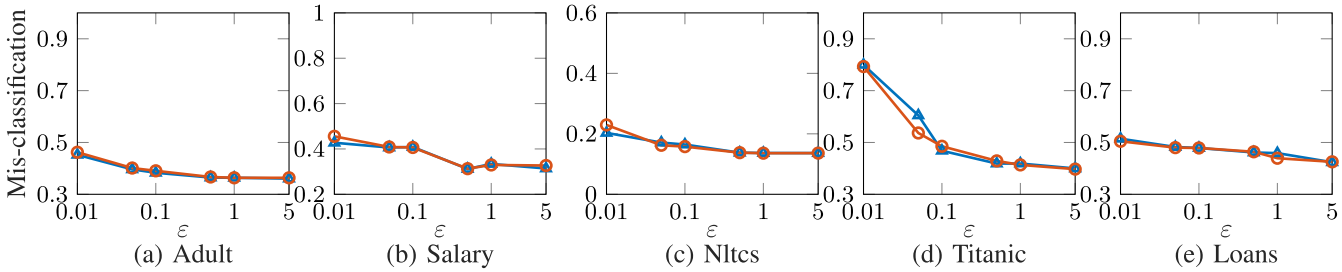


Fig. 9. Mis-classification rate of ABSyn and non-parallel scheme.

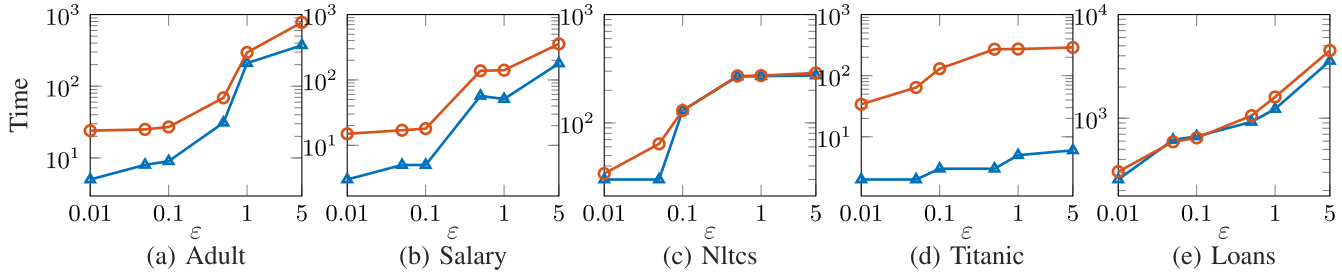


Fig. 10. Running time of ABSyn and non-parallel scheme.

schemes select all marginals under large privacy budgets, i.e., they both train only one clique. The Loans dataset has 50 attributes. The main time overhead of the two schemes is in the select step rather than the generate step, and therefore the differences between the two schemes on this dataset are not significant.

IX. Discussion

In this work, we focus on the accuracy problem of data synthesis and design a practical solution, while some issue still needs to be solved. We give out some future research directions in this area.

A. Optimal Select and Measure Problem

In this paper, we heuristically present an adaptive selection method with our new indicator to provide a better marginal selection result. Our experimental results show that our selection performs well on synthetic data at present. Considering that completing selections in a batch way before the measure step may yield sub-optimal solutions [11], we think it is

important for future research to theoretically prove that some low-sensitivity indicator can approximate the optimal solution.

B. Multiple Data Types

In this paper, we only apply the ABSyn scheme to tabular data, which is not directly applicable to streaming or continuous data. The above two cases must be handled by proper discretization or pre-processing methods, which could impact the quality of synthetic data. Efficient handling of multiple data types without degrading the performance of data synthesis is still an important open problem.

X. Conclusion

In this paper, we propose ABSyn, a scheme for generating differentially private synthetic data. The core innovation of ABSyn lies in the new selection indicator CR-score and the adaptive selection algorithm for batch processes, enabling ABSyn to produce synthetic data with high query accuracy. The CR-score captures high-contribution marginals based on their conditional correlations, while ABSyn's batch processes naturally allocate the optimal privacy budget. In the implementation, our parallel algorithm accelerates generation by training each maximal clique of the graphical model instead of training the graphical model itself. Experimental results demonstrate that ABSyn outperforms prior works in terms of query accuracy.

Differentially private data synthesis is still a popular and challenging research area. We list future research and improvement directions for this paper. A promising direction is to publish a synthetic dataset to represent the dataset held by all parties while protecting their privacy. In the real world, data is commonly held by multiple parties. Data holders can design multi-party secure computation protocols for differentially private data synthesis with utility close to the central model schemes [39], [40], [41]. Additionally, we consider exploring the impact of data heterogeneity on data synthesis. When there is heterogeneity in the datasets of multiple parties, further trade-offs between differential privacy noise and the benefits from independently generating the synthetic dataset for each party are necessary.

References

[1] C. Dwork, "Differential privacy," in *Proc. 33rd Int. Colloq. Automata, Lang., Program.*, Venice, Italy. Berlin, Germany: Springer, Jul. 2006, pp. 1–12.

[2] N. C. Abay, Y. Zhou, M. Kantarcioglu, B. Thuraisingham, and L. Sweeney, "Privacy preserving synthetic data release using deep learning," in *Proc. Eur. Conf.*, Dublin, Ireland. Berlin, Germany: Springer, Sep. 2019, pp. 510–526.

[3] V. Bindschaedler, R. Shokri, and C. A. Gunter, "Plausible deniability for privacy-preserving data synthesis," *Proc. VLDB Endowment*, vol. 10, no. 5, pp. 481–492, Jan. 2017.

[4] A.-S. Charest, "How can we analyze differentially-private synthetic datasets?" *J. Privacy Confidentiality*, vol. 2, no. 2, pp. 21–33, Apr. 2011.

[5] M. Hardt, K. Ligett, and F. McSherry, "A simple and practical algorithm for differentially private data release," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 25, 2012, pp. 1–9.

[6] J. Zhang, G. Cormode, C. M. Procopiuc, D. Srivastava, and X. Xiao, "PrivBayes: Private data release via Bayesian networks," *ACM Trans. Database Syst. (TODS)*, vol. 42, no. 4, pp. 1–41, 2017.

[7] Z. Zhang et al., "PrivSyn: Differentially private data synthesis," in *Proc. 30th USENIX Secur. Symp.*, 2021, pp. 929–946.

[8] R. McKenna, G. Miklau, M. Hay, and A. Machanavajjhala, "Optimizing error of high-dimensional statistical queries under differential privacy," *Proc. VLDB Endowment*, vol. 11, no. 10, pp. 1206–1219, Jun. 2018.

[9] R. McKenna, D. Sheldon, and G. Miklau, "Graphical-model based estimation and inference for differential privacy," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 4435–4444.

[10] R. McKenna, G. Miklau, and D. Sheldon, "Winning the NIST contest: A scalable and general approach to differentially private synthetic data," *J. Privacy Confidentiality*, vol. 11, no. 3, p. 3, Dec. 2021.

[11] R. McKenna, B. Mullins, D. Sheldon, and G. Miklau, "AIM: An adaptive and iterative mechanism for differentially private synthetic data," *Proc. VLDB Endowment*, vol. 15, no. 11, pp. 2599–2612, Jul. 2022.

[12] W. Qardaji, W. Yang, and N. Li, "PriView: Practical differentially private release of marginal contingency tables," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, Jun. 2014, pp. 1435–1446.

[13] D. Chen, T. Orekondy, and M. Fritz, "GS-WGAN: A gradient-sanitized approach for learning differentially private generators," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 12673–12684.

[14] M. Srivastava and M. Alzantot. (2019). *Differentially Private Dataset Release Using Wasserstein GANs*. [Online]. Available: https://github.com/nesl/nist_differential_privacy_synthetic_data_challenge

[15] J. Fan, J. Chen, T. Liu, Y. Shen, G. Li, and X. Du, "Relational data synthesis using generative adversarial networks: A design space exploration," *Proc. VLDB Endowment*, vol. 13, no. 12, pp. 1962–1975, Aug. 2020.

[16] L. Xie, K. Lin, S. Wang, F. Wang, and J. Zhou, "Differentially private generative adversarial network," 2018, *arXiv:1802.06739*.

[17] L. Rosenblatt, X. Liu, S. Pouyanfar, E. de Leon, A. Desai, and J. Allen, "Differentially private synthetic data: Applied evaluations and enhancements," 2020, *arXiv:2011.05537*.

[18] J. Jordon, J. Yoon, and M. Van Der Schaar, "Pate-GAN: Generating synthetic data with differential privacy guarantees," in *Proc. Int. Conf. Learn. Represent.*, 2019, pp. 1–21.

[19] T. Zhu, G. Li, W. Zhou, and P. S. Yu, "Differentially private data publishing and analysis: A survey," *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 8, pp. 1619–1638, Aug. 2017.

[20] S. Aydore et al., "Differentially private query release through adaptive projection," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 457–467.

[21] C. Dwork, M. Naor, O. Reingold, G. N. Rothblum, and S. Vadhan, "On the complexity of differentially private data release: Efficient algorithms and hardness results," in *Proc. 41st Annu. ACM Symp. Theory Comput.*, May 2009, pp. 381–390.

[22] R. Chen, Q. Xiao, Y. Zhang, and J. Xu, "Differentially private high-dimensional data publication via sampling-based inference," in *Proc. Int. Conf. Knowl. Discovery Data Mining*, 2015, pp. 129–138.

[23] R. McKenna, S. Pradhan, D. R. Sheldon, and G. Miklau, "Relaxed marginal consistency for differentially private query answering," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 20696–20707.

[24] M. Gaboardi, E. J. G. Arias, J. Hsu, A. Roth, and Z. S. Wu, "Dual query: Practical private query release for high dimensional data," in *Proc. Int. Conf. Mach. Learn.*, 2014, pp. 1170–1178.

[25] K. Cai, X. Lei, J. Wei, and X. Xiao, "Data synthesis via differentially private Markov random fields," *Proc. VLDB Endowment*, vol. 14, no. 11, pp. 2190–2202, Jul. 2021.

[26] NIST. (2018). *Differential Privacy Synthetic Data Challenge*. [Online]. Available: https://www.nist.gov/ctl/pscr/open-innovation-prize-challenges/past-prize-challenges/2018-differential-privacy-synthetic

[27] NIST. (2020). *Differential Privacy Temporal Map Challenge*. [Online]. Available: https://www.nist.gov/ctl/pscr/open-innovation-prize-challenges/past-prize-challenges/2020-differential-privacy-temporal

[28] F. McSherry and K. Talwar, "Mechanism design via differential privacy," in *Proc. 48th Annu. IEEE Symp. Found. Comput. Sci. (FOCS)*, Oct. 2007, pp. 94–103.

[29] M. Hardt and G. N. Rothblum, "A multiplicative weights mechanism for privacy-preserving data analysis," in *Proc. IEEE 51st Annu. Symp. Found. Comput. Sci.*, Oct. 2010, pp. 61–70.

[30] M. Abadi et al., "Deep learning with differential privacy," in *Proc. ACM SIGSAC Conf. Comput. Commun. Security*, 2016, pp. 308–318.

[31] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Proc. 3rd Theory Cryptography Conf.*, New York, NY, USA. Berlin, Germany: Springer, Mar. 2006, pp. 265–284.

[32] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy," *Found. Trends Theor. Comput. Sci.*, vol. 9, nos. 3–4, pp. 211–407, 2014.

[33] M. Bun and T. Steinke, "Concentrated differential privacy: Simplifications, extensions, and lower bounds," in *Proc. 14th Int. Conf.*, Beijing, China. Berlin, Germany: Springer, Oct. 2016, pp. 635–658.
[34] R. Kohavi, "Scaling up the accuracy of Naive-Bayes classifiers: A decision-tree hybrid," in *Proc. 2nd Int. Conf. Knowl. Discovery Data Mining*, vol. 96, 1996, pp. 202–207.
[35] M. Hay, A. Machanavajjhala, G. Miklau, Y. Chen, and D. Zhang, "Principled evaluation of differentially private algorithms using DPBench," in *Proc. Int. Conf. Manage. Data*, Jun. 2016, pp. 139–154.
[36] K. G. Manton, *National Long-Term Care Survey: 1982, 1984, 1989, 1994, 1999, and 2004*. Ann Arbor, MI, USA: Inter-Univ. Consortium for Political and Social Research, 1982.
[37] F. E. Harrell Jr. and T. Cason. *Titanic Data*. hbiostat.org. Accessed: Sep. 10, 2020. [Online]. Available: https://hbiostat.org/data/repo/titanic
[38] Kaggle. *Kaggle Lending Club Loan Data*. Accessed: Sep. 10, 2020. [Online]. Available: https://github.com/con-fide/ABSyn/blob/main/datasets/loans.csv
[39] S. Maddock, G. Cormode, and C. Maple, "FLAIM: AIM-based synthetic data generation in the federated setting," 2023, *arXiv:2310.03447*.
[40] M. Pereira, S. Pentyala, M. De Cock, A. Nascimento, and R. de Sousa, "Secure multiparty computation for synthetic data generation from distributed data," in *Proc. NeurIPS*, 2022, pp. 1–12.
[41] S. Pentyala, M. Pereira, and M. De Cock, "CaPS: Collaborative and private synthetic data generation from distributed sources," 2024, *arXiv:2402.08614*.

Jingyu Jia received the dual bachelor's degree in information security and law from Nankai University, Tianjin, China, in 2019, where he is currently pursuing the Ph.D. degree in computer science. His research interests include differential privacy and data privacy protection.

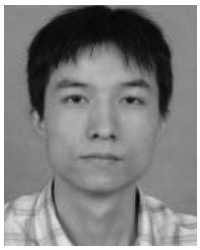
Xinhao Li received the dual bachelor's degree in information management and information systems from Inner Mongolia University, Hohhot, China, in 2022. Currently, he is pursuing the master's degree in computer science with Nankai University. His research interests include differential privacy and data privacy protection.

Tong Li received the B.S. degree in computer science and technology from Taiyuan University of Technology in 2011, the M.S. degree in computer science and technology from Beijing University of Technology in 2014, and the Ph.D. degree in information security from Nankai University in 2017. Currently, he works with Nankai University as an Associate Professor. His research interests include applied cryptography, privacy-preserving computation, and secure machine learning.

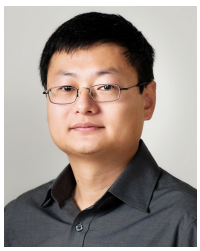
Zhewei Liu received the bachelor's degree in data science and big data technology from China University of Geosciences, Wuhan, China, in 2022. Currently, he is pursuing the master's degree in cyberspace security with Nankai University. His research interests include differential privacy and data privacy protection.

Chang Tan received the B.S. degree in information security and the M.S. degree in computer science from Chongqing University of Posts and Telecommunications, Chongqing, China, in 2017 and 2020, respectively. Currently, he is pursuing the Ph.D. degree in computer science with the College of Computer Science, Nankai University, Tianjin, China. His research interests include differential privacy and data privacy protection.

Siyi Lv received the bachelor's degree in information security and law and the Ph.D. degree in computer science from Nankai University, Tianjin, China, in 2022. Currently, she is a Post-Doctoral Fellow with Nankai University. Her research interests include applied cryptography and data privacy protection.



Liang Guo received the M.Sc. and Ph.D. degrees from Beijing University of Technology, Beijing, China. He joined Huawei Technologies Company Ltd., in 2016. Currently, he works with Huawei as a Chief Engineer. His current research interests include data privacy protection and database security.



Changyu Dong (Member, IEEE) received the Ph.D. degree from Imperial College London. He is currently a Professor with the Institute of Artificial Intelligence and Blockchain, Guangzhou University. He has authored more than 70 publications in international journals and conferences. His recent work focuses mostly on designing practical secure computation protocols. The application domains include secure cloud computing and privacy-preserving data mining. His research interests include applied cryptography, trust management, data privacy, and security policies.

Zheli Liu received the B.Sc. and M.Sc. degrees in computer science and the Ph.D. degree in computer application from Jilin University, China, in 2002, 2005, and 2009, respectively. After a Post-Doctoral Fellowship with Nankai University, he joined the College of Computer and Control Engineering, Nankai University, in 2011. Currently, he works with Nankai University as a Professor. His current research interests include applied cryptography and data privacy protection.