DP-Select: Improving Utility and Privacy in Tabular Data Synthesis with Differentially Private Generative Adversarial Networks and Differentially Private Selection

by

Faezeh Ebrahimianghazani

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Mathematics
in
Computer Science

Waterloo, Ontario, Canada, 2023

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

This thesis proposes DP-Select, a novel approach to tabular data synthesis that combines DP-GAN and differentially private selection. We develop a mutual information-based selection method that is flexible and scalable for high-dimensional data and large numbers of marginals while being differentially private. We evaluate DP-Select on various datasets and demonstrate its effectiveness and utility compared to existing DP-GAN methods. Our results indicate that DP-Select significantly enhances the utility of synthesized data while maintaining strong privacy guarantees, making it a promising extension of DP-GANs for privacy-preserving data synthesis in terms of differential privacy. We also show that DP-Select performs better for smaller privacy budgets, making it an attractive option for scenarios with limited privacy budgets.

Acknowledgements

I would like to express my sincere gratitude to Professor Kerschbaum for his invaluable guidance, support, and feedback throughout the development of this thesis. His expertise, wisdom, and mentorship have been instrumental in shaping my academic and research journey. I would also like to extend my thanks to all the esteemed faculty members and staff at the CrySP lab for their valuable insights, discussions, and contributions to my growth as a researcher. Lastly, I am grateful to my dear fellow graduate students for their camaraderie, encouragement, and collaboration during this enriching experience.

Dedication

To my beloved family and dear friends, who have always been my constant source of love, support, and encouragement, I dedicate this thesis. Your unwavering belief in me has been my strength throughout this journey, and I could not have done it without you.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Sharing high-dimensional and sensitive datasets, particularly in fields such as healthcare and finance, presents a challenge due to privacy concerns and regulatory limitations. Traditional anonymization techniques are often insufficient to protect privacy [29], necessitating novel approaches to preserve the usefulness of data while maintaining privacy. Synthetic data, generated by models, provides a solution by allowing the user to control the amount of private information released and the resemblance to real data. Synthetic data is a useful substitute for real data in situations where data privacy is a concern, while still enabling analysis and learning [30]. Our work focuses on improving the utility and privacy of synthetic data generation through a novel approach that builds on differentially private generative adversarial networks and differentially private selection.

Differential privacy is a rigorous framework for protecting sensitive information from malicious actors, and it has become a widely accepted standard for privacy preservation [17]. This framework provides strong theoretical privacy guarantees, which have been adopted by researchers and industry leaders (e.g., [3, 13, 45]). Specifically, a randomized algorithm satisfies differential privacy if for any two neighboring datasets (i.e., datasets that differ by the inclusion or exclusion of a single data point), the probability of producing any particular output is nearly the same, up to a factor of at most $\exp(\epsilon)$, where ϵ is a privacy parameter. In other words, the distance between the output distributions of the algorithm for two input datasets, where one dataset has a single additional datapoint, is bounded.

There are different approaches for differentially private data synthesis. DPSGD (differentially private stochastic gradient descent) [1] is a crucial component in many deep learning-based algorithms that enables differentially private training of neural networks.

Although DPSGD was not originally designed for generative models, it can be applied to both GANs [49, 54] and VAEs [10]. Another method that is commonly used is the PATE mechanism [38], which can create a private predictor for black-box models. In GAN models, the discriminator has been replaced with a private PATE model, and it has also been used as a way to pass gradients from discriminator to generator privately [52]. Additionally, some approaches represent data in a low-dimensional form, such as using a Bayesian network [53] to learn the generation process. The NIST competition[1] was won with an algorithm that learns all 2-way distributions in a differentially private way and generates data from potentially inconsistent marginals through post-processing [33].

Fan et al. (2020) [22] presents a detailed experimental study on the application of GAN to relational data synthesis. The paper reveals that the current differential privacy (DP) preserving GAN solutions are less effective than traditional data synthesis methods that ensure DP. While GAN approaches for data synthesis are still subject to ongoing research [24, 56], this thesis aims to enhance the current differentially private GAN methods by introducing a postprocessing step. The proposed DP-Select algorithm, in combination with GAN, has the potential to become a competitive data synthesis approach that ensures both privacy and utility.

In this thesis, we propose a new approach called DP-Select that improves the utility and privacy of tabular data synthesis. DP-Select builds upon a differentially private generative adversarial network (DP-GAN) to generate a pool of potential data points that are similar to the original dataset. We then use a differentially private selection method based on mutual information to select the most representative data points that match a set of selected marginals.

We perform experiments to assess the efficacy of DP-Select on two datasets and in various privacy regimes. Our findings demonstrate that DP-Select improves the utility of synthesized data considerably when compared to DP-GAN alone, as assessed by classification accuracy. DP-Select attains higher levels of utility while ensuring robust privacy guarantees, rendering it a promising approach for differential privacy-based privacy-preserving data synthesis. Furthermore, our experiments revealed that DP-Select's improvement over DP-GAN is more significant with lower privacy budgets compared to higher privacy budgets, making it an appealing alternative for scenarios with constrained privacy budgets. However, DP-Select is a very inefficient algorithm requiring considerable computational resources. In order to scale DP-Select to larger data sets we propose a parallel version of the algorithm that leverages parallel composition in differential privacy. Through exper-

[1] https://www.nist.gov/ctl/pscr/open-innovation-prize-challenges/past-prize-challenges/2018-differential-privacy-synthetic

imentation, we found that using five parallel threads, our algorithm achieves maximum utility while reducing the running time to one-seventh of the original algorithm.

In summary, the contributions of this thesis are:

- The introduction of DP-Select, a new technique for generating tabular data that combines DP-GAN with differentially private selection.
- The design of a differential privacy-based selection method that utilizes mutual information and is adaptable to high-dimensional data and numerous marginals.
- The development of a parallel version of DP-Select that significantly improves its efficiency while maintaining or even improving its utility, making it scalable to larger datasets.
- An empirical evaluation of DP-Select on various datasets that demonstrates its effectiveness and utility compared to existing methods.

Chapter 2

Related Work

Differential privacy (DP) is a well-established and widely accepted concept for privacy protection and a state-of-the-art approach for ensuring privacy in data analysis [17]. Differential privacy has received significant attention due to its ability to provide provable privacy guarantees and quantifiable privacy loss, as well as offering strong protection against re-identification and re-construction attacks [20]. Differential privacy can be ensured by integrating it into non-private algorithms, such as those designed for specific tasks such as classification or statistical data release, in order to safeguard the individuals' information used as input. However, this requires redesigning each algorithm individually. In contrast, this thesis focuses on generating a differentially private synthetic dataset that can serve as input for existing non-private algorithms. Although this approach may not optimize utility for a specific task, it is more versatile since any task can be performed using the synthetic dataset without the need to modify existing non-private algorithms. A number of previous studies have been carried out on generating differentially private synthetic datasets (e.g., [49, 52, 40]), and this thesis contributes to this body of work.

2.1 Deep generative methods

DPGAN by Xie et al. (2018) [49] is a differentially private generative adversarial network that uses moments accountant [2] to ensure the privacy of sensitive data while generating synthetic data that closely resemble the original data. DPGAN uses DPSGD [1] and adopts the WGAN [4] objective. DPGAN clips the model weights w, to ensure the discriminator network is Lipschitz [4]. Authors show that by clipping w to a bounded box, the gradients

are automatically bounded by some constant c_g, without an explicit gradient clipping step as in DPSGD.

PATE-GAN, proposed by Jordan et al. (2019) [52], employs the PATE method[38, 39] for differential privacy. It trains the student discriminator using generated samples that are labeled by the teachers and do not require the student discriminator to access publicly available datasets.

DP-CGAN by Torkzadehmahani et al. (2019) [47] adopts the CGAN [37] objective that allows both the generator and discriminator to be conditional on some side information such as the class label. DP-CGAN uses Renyi Differential Privacy (RDP) [36] accountant to obtain a tighter estimation on the differential privacy guarantees compared to moments accountant [2]. Several recent approaches concentrate on particular scenarios, such as decentralized databases. DP-FedAvg-GAN by Augenstein et al. (2019) [5] proposes to train differentially private generative models with federated learning.

DPGAN [49] and PATE-GAN [52], among the methods discussed above, can be applied to tabular data. However, the unique properties of tabular data, such as correlated features, mixed data types, and potential mode collapse, pose difficulties for GANs to learn the tabular data distribution. Therefore, there exist differential privacy generative models that are tailored specifically for tabular data. Table-GAN, a GAN-based architecture developed by Park et al. (2018) [40], represents one of the initial attempts to tackle privacy concerns by generating synthetic tabular data that exhibit comparable statistical properties to the original table. DP-CTGAN by Fang et al. (2022) [24] adapts the CTGAN model [50] to generate secure tabular medical data in the federated learning setting. CTAB-GAN, proposed by Zhao et al. (2021) [56], is designed to address data imbalance and long-tail issues while effectively modeling diverse data types, such as a mix of continuous and categorical variables.

There are several works available that survey and compare existing differentially private data synthesis methods [23, 46], but only a few of them specifically focus on differentially private GANs for tabular data synthesis. One of the works that particularly motivated this thesis is published by Fan et al. in 2020 [22]. In this paper, the authors conducted a comprehensive experimental study for applying GAN to relational data synthesis. The authors demonstrate that the current solution for differential privacy (DP) preserving GAN is not as effective as traditional data synthesis methods that offer DP guarantees. Therefore, the goal of this thesis is to improve the existing differentially private GAN methods by adding a postprocessing step. To avoid complexity, we have opted to use the simple DPGAN [49] as our base model.

2.2 Other methods

In this subsection, we explore non-GAN methods for producing differentially private synthetic datasets, which can be classified into two categories: game-based methods and graphical model-based methods.

MWEM [27] and Dual Query [25] are two approaches that view dataset synthesis as a two-player zero-sum game. MWEM maintains the distribution of the data player using a no-regret algorithm. However, when the dataset domain is large, it becomes infeasible to maintain the full distribution. On the other hand, Dual Query maintains a distribution over queries instead of the dataset, with each generated record consuming a portion of the privacy budget. Both methods require a predetermined workload of queries, rendering them unsuitable for handling arbitrary kinds of tasks with sufficient accuracy. Although recent works [48] have improved MWEM and DualQuery by replacing their core components, they still rely on the exponential mechanism to provide privacy and do not address the fundamental limitations of these methods.

Graphical Model Based Methods (GMBMs) aim to estimate a graphical model that provides an approximation of the distribution of the original dataset in a differentially private manner. Some GMBMs, such as PrivBayes [53] and BSG [7], employ a Bayesian Network to approximate the data distribution. PrivBayes initially employs a private process to determine the network structure, followed by obtaining the noisy conditional probability distribution of each node. Other GMBMs, such as PGM [34] and JTree [11], utilize Markov Random Field to approximate the data distribution. PGM aims to estimate a Markov Random Field that best matches a set of predefined low-dimensional marginals, while JTree aims to estimate a dependency graph and subsequently transform it into a junction tree to obtain the Markov Random Field. However, a primary limitation of GMBMs is their inability to handle dense marginals that capture more correlation information.

PrivSyn [55] proposes a different approach to dataset representation, using a large set of low-degree marginals instead of graphical models. The authors [55] introduce a metric, called InDif, to privately measure the correlation between pairwise attributes and select the marginals based on these measurements. They add guassian noise to the selected marginals to ensure privacy. Additionally, PrivSyn iteratively updates a synthetic dataset to ensure it matches the target set of noisy marginals. While PrivSyn does not use Mutual Information (MI) due to its high global sensitivity, we can leverage MI in DP-Select because we do not calculate the marginals privately. Rather, we use marginals privately to iteratively update the synthetic dataset.

One major advantage of our method, DP-Select, is that it builds upon a DP-GAN,

which already generates a pool of potential data points similar to the original dataset at a cost of a part of the privacy budget $\epsilon_{dp\text{-}gan}$. Furthermore, our selection method is not limited to a specific structure, such as graphical models, and is, therefore, more flexible and adaptable to a wider range of datasets. Additionally, since we do not compute the marginals privately and the sensitivity of the score function is not dependent on the number of marginals, our algorithm is not limited by the number or dimension of the marginals except for computation limitations. This makes DP-Select a highly scalable and versatile approach to dataset synthesis.

Chapter 3

Background and Problem Statement

In this section, we introduce the fundamental concepts of differential privacy and briefly review the Generative Adversarial Network (GAN) architecture and its differentially private variant, DP-GAN. Finally, we present a formal problem definition for the task of differentially private data synthesis with DP-Select.

3.1 Differential Privacy

Differential privacy [18] is the privacy model used in our approach, which aims to protect the privacy of sensitive input data. A randomized algorithm M is said to satisfy differential privacy if the presence of a data point in the input to M cannot be distinguished by the output of M, except with a bounded probability.

Definition 1. ***Differential Privacy [20].*** *A randomized algorithm M is (ϵ, δ)-differentially private if for any two neighboring datasets D and D' (differing in a single point) and for any subset of outputs S:*

$P(M(D) \in S) \leq e^{\epsilon} P(M(D') \in S) + \delta,$

where $M(D)$ and $M(D')$ are the outputs of the algorithm for neighboring datasets D and D', respectively.

It can be shown that the definition is equivalent to:

$|\log \frac{(P(M(D)=s))}{(P(M(D')=s))}| \leq \epsilon,$

with probability at least $1 - \delta$ for any two neighboring datasets D and D', where ϵ reflects the privacy level. A small ϵ implies small differences in output probabilities between databases D and D', indicating high perturbations and therefore high privacy.

One of the fundamental properties of differential privacy is post-processing. This means that if the output of a differentially private mechanism is further processed by a function that does not depend on the input data, the resulting output remains differentially private.

Lemma 1. ***Post-Processing.*** *If $M(x)$ satisfies (ϵ, δ)-differential privacy, then for any (deterministic or randomized) function g that is independent of x, $g(M(x))$ satisfies (ϵ, δ)-differential privacy.*

3.2 Generative Adversarial Network

A Generative Adversarial Network (GAN) [26] is a deep learning framework that learns to generate samples from a target distribution $p_{data}(\mathbf{x})$ by training two neural networks: a generator G and a discriminator D. The generator maps a latent vector $\mathbf{z}$ from a prior distribution $p_{\mathbf{z}}(\mathbf{z})$ to a sample $\mathbf{x} = G(\mathbf{z})$, while the discriminator tries to distinguish between samples from the true data distribution and those generated by the generator. The two networks are trained simultaneously in a two-player minimax game, where the objective of the generator is to fool the discriminator into thinking that its samples are real, and the objective of the discriminator is to correctly distinguish between real and fake samples. The training process can be formalized as follows:

$$\min_G \max_D \{\mathbb{E}_{\mathbf{x} \sim p_{data}(\mathbf{x})}[\log D(\mathbf{x})] + \mathbb{E}_{\mathbf{z} \sim p_{\mathbf{z}}(\mathbf{z})}[\log(1 - D(G(\mathbf{z})))\}, \tag{3.1}$$

where D and G are the discriminator and generator functions, respectively, and the objective function is the binary cross-entropy loss.

3.2.1 Differentially Private Generative Adversarial Network

DP-GAN [49] is a variant of GAN that satisfies differential privacy by adding noise to the gradients during the training process and by clipping the gradients to a maximum L2 norm [1]. Specifically, DP-GAN adds Gaussian noise with variance σ^2 to the gradients of the discriminator during each iteration of training, where σ is chosen such that the resulting mechanism satisfies a given privacy budget ϵ. The privacy guarantee is achieved

Denotation List	
$\mathcal{D}$	the original dataset
$N_{dp\text{-}gan}$	a DP-GAN trained on $\mathcal{D}$
G	the generator function of $N_{dp\text{-}gan}$
$Pool$	a pool of candidate synthetic data generated by G
$\mathcal{D}'$	the final output synthetic dataset

Table 3.1: A table of denotations used in this thesis.

by using the moments accountant technique [2] to keep track of the privacy cost of the noise added at each iteration. The noise addition has the effect of smoothing the output of the discriminator, which in turn makes it more difficult for an attacker to infer sensitive information about the input data.

3.3 Problem Definition

The problem we aim to address is the low utility of differentially private generative adversarial networks (DP-GANs), as reported in previous research [22]. To improve the utility of DP-GANs, we propose a post-processing approach using differentially private selection. Specifically, we consider the problem of generating a differentially private synthetic dataset $\mathcal{D}'$ that closely approximates the distribution of an original dataset $\mathcal{D}$ while preserving privacy. Our approach assumes that we have access to a trained DP-GAN, denoted as $N_{dp\text{-}gan}$, which is trained on $\mathcal{D}$ with a privacy parameter $\epsilon_{dp\text{-}gan}$. Using $N_{dp\text{-}gan}$'s generator G, we generate a large number of candidate synthetic data, denoted as $Pool$. Our goal is to select a subset of $Pool$ that preserves the utility of $\mathcal{D}$, meaning that a classifier trained on $\mathcal{D}'$ performs comparably to one trained on $\mathcal{D}$, and satisfies a differential privacy constraint with a given privacy budget $\epsilon_{dp\text{-}select}$. Therefore, the research question we address is whether post-processing DP-GANs using selection can improve their utility.

3.3.1 Utility

The utility of synthetic data depends on the intended use in downstream applications. In this thesis, we focus on the use of the synthesized dataset for training machine learning (ML) models, which has been a common focus in recent works [22, 9, 50, 51, 40, 6, 12, 31]. In other words, the machine learning model trained on the synthetic dataset $\mathcal{D}'$ should

yield comparable performance to that trained on the original dataset $\mathcal{D}$. To streamline the evaluation process, this thesis focuses on classification models, specifically decision trees.

Chapter 4

Algorithm

This thesis proposes DP-Select, a post-processing selection algorithm to improve the utility of data synthesized by DP-GAN. The algorithm selects a subset of synthetic data generated by the DP-GAN based on their similarity to the marginals of the original dataset. Mutual information is used to identify the important marginals, and an iterative process is used to add a new data point in each iteration to the output dataset while ensuring privacy with the exponential mechanism.

DP-Select is effective at preserving differential privacy but can face performance issues with large datasets for two reasons. First, computing the marginals of potential data points becomes increasingly time-consuming as the dataset grows, leading to longer running times. Second, the privacy budget for each iteration of the exponential mechanism decreases with large datasets, resulting in low utility. These performance issues can make DP-Select impractical for real-world applications where fast and efficient data processing is necessary. To address these performance issues, we propose Parallel DP-Select in Section 4.4.

4.1 Overview of the Algorithm

The algorithm starts by training a DP-GAN on a private dataset, producing a synthetic dataset as output. We use the generator of the DP-GAN to create a semi-infinite pool of synthetic data, which has 10-20 times larger size than the original dataset.

We then select a subset of this semi-infinite pool as the output dataset. To do that we first use mutual information to calculate the most important low-dimensional marginals for the original dataset. In the iterative process, we select a datapoint from the pool

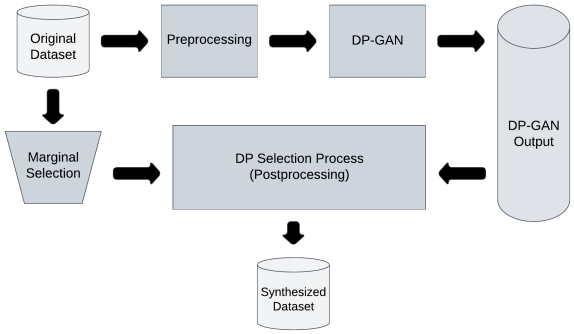

Figure 4.1: Overview of the algorithm

that maximizes the similarity of the output dataset to the original dataset in terms of the marginals we have selected. We use the exponential mechanism to protect privacy by randomizing the selection process, ensuring that the marginals that were derived directly from the original dataset maintain private.

We repeat the iterative process until the desired number of data points have been selected. The final output dataset is the selected subset from the pool. Figure 4.1 shows a summary of the algorithm.

4.2 Training the DPGAN

Generative adversarial networks (GANs)[26] are a class of machine learning models that consist of two components: a generator and a discriminator. The generator learns to produce synthetic data samples that resemble real data, while the discriminator learns to distinguish between real and synthetic data samples. The two components are trained together in a process called adversarial training.

In an adversarial process, the generator tries to produce synthetic samples that fool the discriminator, while the discriminator tries to correctly identify which samples are real and which are synthetic. The objective of the generator is to maximize the probability that the discriminator incorrectly labels a synthetic sample as real. The objective of the discriminator is to correctly classify the samples as real or synthetic.

By training the generator and discriminator in this way, we can create a model that can produce synthetic data samples that closely resemble real data. Adding noise to

the gradient updates during training and clipping the gradients of the discriminator are techniques used to ensure differential privacy [1].

To train the DP-GAN, we follow the standard DPGAN[49] and use a standard WGAN (Wasserstein GAN)[4] architecture with added noise and clipping the gradients of the Discriminator to ensure differential privacy. Through the moment accountant mechanism[2], we can compute the final composition result ϵ.

Lemma 2. *[49] The output of generator learned in a Differentially Private Generative Adversarial Network guarantees (ϵ, δ)-differential privacy.*

Once the DP-GAN is trained, the generator can be used to create a semi-infinite pool of synthetic data samples. We use this pool as input to our iterative process for improving the synthesized data, as described in the next section.

4.3 DP-Selection process

The purpose of the algorithm is to select the best subset of a large pool of potential data points that match the original dataset in some way, such as marginal distributions. The algorithm proceeds as follows:

1. Randomly select some potential data points from the pool.
2. Calculate a score for each of the selected data points, which reflects how adding each data point to the output dataset would affect the similarity between the output dataset and the original dataset. The score is based on the distance between the marginal distributions of the original dataset and the potential output dataset.
3. To provide privacy, the algorithm uses the exponential mechanism to select the data point with the highest score and adds it to the output dataset.
4. Repeat steps 1-3 until the output dataset size is equal to the desired size of the output dataset.

To balance runtime and utility performance, we randomly select T potential data points from the pool as the first step. T can be equal to the size of the dataset pool, but this can result in slow runtime without significant improvement in utility. Alternatively, choosing a smaller T may result in a noticeable increase in runtime but a slight decrease in utility.

Algorithm 1: DP-Select

Input: $\mathcal{D}$: Original dataset
ϵ_{total}: total privacy budget used
ϵ-$ratio$: ratio of $\epsilon_{dp\text{-}gan}$ to ϵ_{total}

Result: Selected: a subset of potential data-points selected for output dataset

$\epsilon_{dp\text{-}gan} \leftarrow \epsilon\text{-}ratio \times \epsilon_{total}$; `/* initialization */`

$\epsilon_{dp\text{-}select} \leftarrow (1 - \epsilon\text{-}ratio) \times \epsilon_{total}$;

$pool\text{-}size \leftarrow 10 \times |\mathcal{D}|$;

$selected\text{-}size \leftarrow |\mathcal{D}|$;

$T \leftarrow 20$; `/* number of random data points selected in step 1 */`

$Pool \leftarrow$ **DP-GAN**$(\mathcal{D}, \epsilon_{dp\text{-}gan}, pool\text{-}size)$;

; `/* generating a pool of potential data points */`

$\epsilon_{partial} \leftarrow Bounded\text{-}range\text{-}composition(\epsilon_{dp\text{-}select}, |\text{Selected}|)$;

; `/* calculation of the privacy budget used in each iteration */`

Selected $= \{\}$;

while $|\text{Selected}| < selected\text{-}size$ **do**

  $potential \leftarrow rand(Pool, T)$; `/* step 1 */`

  $scores_i \leftarrow score\text{-}function(\text{Selected} \cup potential_i, \mathcal{D})$; `/* step 2 */`

  $selected\text{-}datapoint \leftarrow exponential\text{-}mechanism(potential, scores, \epsilon_{partial})$;
  `/* step 3 */`

  Selected $\leftarrow$ Selected $\cup$ $selected\text{-}datapoint$

end

In our experiments, we found that selecting $T = N$ is not efficient due to poor runtime and insignificant utility improvement. With $T = m \times \frac{|Pool|}{|\mathcal{D}|}$, the probability of a set of c datapoints not being selected randomly in the whole algorithm is less than e^{-mc}. Moreover, the order of adding data points to the output has a low impact on output dataset utility. Therefore, selecting a smaller T with negligible utility loss is acceptable. We use $T = \frac{|Pool|}{|\mathcal{D}|}$ in our experiments.

The algorithm's approach balances utility and privacy by choosing the data points with the highest scores to maintain similarity to the original dataset while protecting privacy through the exponential mechanism. This technique ensures that the output dataset mimics the marginal distributions of the original dataset, providing an effective solution for privacy-preserving data selection.

In the following subsections, we will provide more details on each step of the algorithm process, including the selection of important marginal distributions, the calculation of the score for each data point, and the application of the exponential mechanism to ensure privacy. You can find a pseudo-code that outlines the steps of our DP-selection algorithm in Algorithm 1. This provides a clearer understanding of how the algorithm works and how it can be implemented.

4.3.1 Marginal selection

To ensure that the synthetic dataset preserves the relationships between the features of the original dataset, we select the k most important marginals using mutual information. This helps to establish a reliable basis for comparing the synthetic and original datasets. Specifically, we calculate the mutual information between each pair/triplet of features and select the k highest ones. The mutual information between two random variables X and Y is defined as:

$I(X;Y) = \sum_{y \in Y} \sum_{x \in X} p(x,y) \log \frac{p(x,y)}{p(x)p(y)}$

where $p(x,y)$ is the joint probability distribution of X and Y, and $p(x)$ and $p(y)$ are their marginal probability distributions.

After selecting the k most important marginals, we also calculate their importance weights based on the mutual information values. Specifically, the importance weight of each marginal is calculated as:

$w_i = \frac{I_i}{\sum_{j=1}^{k} I_j}$

where I_i is the mutual information value of the *i-th* marginal, and w_i is its corresponding importance weight.

The value of k is chosen based on the desired level of accuracy and the available computational resources. In our experiments, we found that selecting the top 10% of the marginals resulted in a good balance between accuracy and computational efficiency.

By computing the distance to the k most informative marginals selected based on mutual information, we ensure that the synthetic dataset generated by our algorithm maintains the relationships between the features of the original dataset, thereby producing similar marginal distributions.

4.3.2 Score function

Algorithm 2: Score function

Input: $\mathcal{D}$: Original dataset
$\hat{\mathcal{D}}$: a subset of potential data points selected for output dataset until now
c: the candidate data point
Result: *score*: score of the candidate point
for *attr-pair* $= (attr_1, attr_2)$ **where** $attr_1, attr_2 \in \mathcal{D}.attributes$ **do**
  $X, Y \leftarrow \mathcal{D}_{attr1}, \mathcal{D}_{attr2}$;
  $I(\textit{attr-pair}) \leftarrow \sum_{y \in Y} \sum_{x \in X} p(x,y) \log \frac{p(x,y)}{p(x)p(y)}$; `/* mutual information between two attributs */`
end
best-marginals $\leftarrow$ *k attr-pairs with k highest* $I(\textit{attr-pair})$;
$weight_i \leftarrow \frac{I_i}{\sum_{j=1}^{k} I_j}$ *for* $i \in \{1, ..., k\}$;
$\mathcal{D}_{temp} \leftarrow \hat{\mathcal{D}} \cup \{c\}$;
score $\leftarrow 0$;
for *marginal* $\in$ *best-marginals* **do**
  $score \leftarrow score + weight_i \times Dist(\mathcal{D}, \mathcal{D}_{temp}, marginal)$
end
return $-score$;

To evaluate the similarity between the output and original datasets, we calculate the distance between their marginal distributions using three distance measures: Density distance area, Kolmogorov distance, and L_2 density distance.

Definition 2. ***Kolmogorov distance.*** *The Kolmogorov distance is a statistical measure of the difference between two probability distributions. It is defined as the maximum absolute difference between the cumulative distribution functions (CDFs) of the two distributions. More formally, given two cumulative distribution functions F and G respectively for two distributions X and Y, the Kolmogorov distance is defined as:*

$$D_K(X, Y) = \sup_x |F(x) - G(x)|$$

where x is a value in the support of both distributions and $\sup_x$ denotes the supremum (i.e., the least upper bound) over all possible values of x. The Kolmogorov distance is a non-negative real number between 0 and 1, with 0 indicating that the two distributions are identical, and 1 indicating that they are completely different.

Definition 3. ***Density distance area [8].*** *The density distance, also known as the density difference area, is a measure of distance between two probability density functions (PDF). If X and Y have PDFs f and g, the density distance between X and Y is defined as:*

$$D_D(X, Y) = \int |f(x) - g(x)|dx$$

where the integral is taken over the entire domain of x. If X and Y are discrete random variables taking integer values, the density distance can be expressed as:

$$D_D(X, Y) = \sum_{i=1}^{n} |f(X = x_i) - g(Y = x_i)|$$

where the summation is taken over all possible values of x_i.

$D_D(X, Y)$ is the L_1 distance between the PDFs of X and Y.

Definition 4. L_2 ***density distance.*** *The L2 density distance is a measure of distance between two probability density functions (PDFs), denoted by f and g, based on the L_2 norm of their difference. It is defined as:*

$$D_{L2}(X, Y) = \sqrt{\int (f(x) - g(x))^2 dx}$$

This formula is similar to the Density Distance Area, but instead of taking the absolute value of the difference between the densities, it squares the difference and takes the square root of the integral of the result.

Based on experiments, we found that the density distance measure yields the best results. Density distance measure computes the distance between the histogram of the marginal distributions of the two datasets by comparing the areas between their density curves.

To calculate the score of a candidate data point, we temporarily add it to the output dataset and calculate the distance between its marginal distributions and those of the original dataset using the Density distance measure. Since some marginal distributions may be more important than others in preserving the relationships between the features, we weigh the distances using the importance weights calculated in the previous step. Finally, we sum up the weighted distances of all marginal distributions to obtain the distance score of the candidate data point. Since we need to select the candidate with the lowest distance score, we negate the score and use it as the input to the exponential mechanism. The mechanism is designed to output an approximate $argmax$ of the input function, so we select the candidate with the lowest distance (negated) score. The score function can be found in Algorithm 2.

Lemma 3. *The global sensitivity of the score function with density distance area as the distance metric is* $\frac{2}{|D|+1}$*, where* $|D|$ *is the size of the dataset.*

Proof. The sensitivity of the score function is defined as:

$$\Delta_s = \max_{\mathcal{D},\mathcal{D}':d(\mathcal{D},\mathcal{D}')\leq 1} |score(\mathcal{D}) - score(\mathcal{D}')|, \tag{4.1}$$

where $d(\mathcal{D}, \mathcal{D}')$ represents the distance between two datasets $\mathcal{D}$ and $\mathcal{D}'$, and we say that two datasets are neighbors if their distance is 1 or less.

$\{M_1^D, M_2^D, ..., M_k^D\}$ denote the k important marginals of dataset $\mathcal{D}$. All marginals are normalized and each M_i^D has b bins. We denote the value of j^{th} bin of M_i^D by $M_i^D(j)$.

If dataset $\mathcal{D}'$ has one data point x_0 added to the dataset $\mathcal{D}$ and x_0 belongs to the j_0^{th} bin of M_i^D, then the marginal M_i^D of $\mathcal{D}'$ differs from $M_i^{D'}$ as follows:

$$M_i^{D'}(j) = \begin{cases} \frac{M_i^D(j)\times|D|+1}{|D|+1}, & j = j_0. \\ \frac{M_i^D(j)\times|D|}{|D|+1}, & j \neq j_0. \end{cases} \tag{4.2}$$

The maximum value of the sensitivity occurs when the difference between the density distance areas of the marginals and the selected potential data points is maximum. If we assume that the selected potential data points do not change, then the maximum difference in the density distance area is given by:

$$|Dist(\mathcal{D}) - Dist(\mathcal{D}')| < |\frac{M_i^D(j_0) \times |D| + 1}{|D| + 1} - M_i^D(j_0)| + \sum_{j \neq j_0} |M_i^D(j_i) - \frac{M_i^D(j_i) \times |D|}{|D| + 1}|$$
$$= \frac{1 - M_i^D(j_0)}{|D| + 1} + \sum_{j \neq j_0} \frac{M_i^D(j_i)}{|D| + 1} < \frac{1}{|D| + 1} + \frac{\sum_{j \neq j_0} M_i^D(j_i)}{|D| + 1} < \frac{2}{|D| + 1} \quad (4.3)$$

Since the score function is a weighted sum of the distance values, the sensitivity of the score function is $\frac{2}{|D|+1}$.

□

4.3.3 Exponential mechanism

The exponential mechanism, first introduced by Frank McSherry and Kunal Talwar in 2007[35], is a method used to create algorithms that ensure differential privacy. There is a private set domain $\mathcal{D}$ and an object domain $\mathcal{H}$. It maps a set of n inputs from domain $\mathcal{D}$ and an object function $h \in \mathcal{H}$ to a probability distribution over the range $\mathbb{R}$. The privacy mechanism makes no assumption about the nature of $\mathcal{D}$ and $\mathcal{H}$.

Let $s : \mathcal{D}^n \times \mathcal{H} \hookrightarrow \mathbb{R}$ be a function that assigns a score to the pair (X, h), where $X \in \mathcal{R}^n$ and $h \in \mathcal{H}$. The score reflects the appeal of the pair (X, h), i.e., the higher the score, the more appealing the pair is. Given the input $X \in \mathcal{D}^n$, the mechanism's objective is to return an $h \in \mathcal{H}$ such that the function $s(X, h)$ is approximately maximized.

Lemma 4. *The exponential mechanism (M_E) is a mechanism that takes inputs X, H, and s and outputs an object $h \in H$, where the probability of selecting a specific h is proportional to* $\exp(\frac{\epsilon s(X,h)}{2\Delta})$. *The exponential mechanism M_E is epsilon-diferentially private.*

In our process, the input $X \in \mathcal{D}^n$ corresponds to the original dataset that we wish to protect, s is a score function that assigns a score to candidate results based on their desirability, and $\mathcal{H}$ is a set of random candidates generated by the mechanism. The objective of the mechanism is to return a candidate result h such that the score function $s(X, h)$ is approximately maximized, while ensuring that the original dataset remains private.

To achieve this, we use the exponential mechanism $M_E(\mathcal{D}, \mathcal{H}, s)$ as follows:

- Define $s(X, h)$ as the score function that evaluates the desirability of candidate results $h_i \in \mathcal{H}$ given the original dataset as explained in Section 4.3.2.

- Choose h with probability proportional to $\exp(\frac{\epsilon s(X,h)}{2\Delta})$, where ϵ is a privacy parameter that determines the level of privacy protection, and Δ is the sensitivity of the score function.

By setting ϵ to a suitably small value, we can ensure that the mechanism provides strong privacy protection for the original dataset, while still returning a desirable candidate result h.

4.3.4 Composition theorems and bounded-range DP

One of the crucial properties of differential privacy is its behavior under composition. When running multiple differentially private algorithms on the same dataset, the resulting composed algorithm is also differentially private, but with some degradation in the privacy parameters (ϵ, δ). This makes composition a useful tool in algorithm design, as it allows for the combination of simpler algorithms to create new differentially private algorithms. However, the privacy cost of each individual query may accumulate, leading to a higher overall privacy cost for the combined queries. Composition theorems provide bounds on the overall privacy cost of multiple queries based on the individual privacy guarantees of each query.

Formally, we define the composition of differentially private mechanisms $M_0, M_1, ..., M_{k\text{-}1}$ as $M = Comp(M_0, M_1, ..., M_{k\text{-}1})$, where each M_i is run independently. The Sequential Composition Theorem[19] states that the privacy degradation is at most linear with the number of mechanisms executed ($k\epsilon$), while the Advanced Composition Theorem[21] allows for a proportional degradation of ϵ to the square root of the number of mechanisms executed ($\sqrt{k}\epsilon$), with an increase in δ.

Lemma 5. *If F_0 satisfies ϵ_0-differential privacy And F_1 satisfies ϵ_1-differential privacy, then the mechanism (F_0, F_1) which releases both results satisfies $(\epsilon_0{+}\epsilon_1)$-differential privacy.*

For specific differential privacy mechanisms, further composition bounds can be obtained through tighter analysis. In the case of our algorithm, we use the exponential mechanism repeatedly in the DP-selection part, which has a tighter analysis under bounded range DP[16].

Definition 5. ***Bounded Range DP.*** *If a mechanism M transforms a set of records in D into an outcome set R, then we define M as ϵ-range-bounded (ϵ-BR) if for every $y, y' \in R$ and any neighboring databases D and D', the following condition holds:*

$\frac{Pr[M(D)=y]}{Pr[M(D')=y]} \leq e^{\epsilon} \frac{Pr[M(D)=y']}{Pr[M(D')=y']}$.

Bounded range DP is a general notion of privacy, but it is particularly useful for exponential mechanisms. All ϵ-DP mechanisms satisfy 2ϵ-BR, with the exponential mechanism enjoying a tighter analysis as ϵ-BR[16].

The tighter analysis of ϵ-BR mechanisms allows for the derivation of tighter composition bounds, as demonstrated by Durfee and Rogers in their work on top-k queries[16], which was later improved by Dong et al.[14]. Although the optimal bound for the composition of ϵ-BR mechanisms does not have a closed-form expression, a preliminary result with a closed-form expression was proven by computing the supremum of the KL divergence.

Lemma 6. *The adaptive composition of a ϵ_s-BR mechanism under n-fold adaptive composition is (ϵ_t, δ)-DP with:*

$$\epsilon_t = min\{n\epsilon_s, n(\tfrac{\epsilon_s}{1-e^{-\epsilon_s}} - 1 - ln(\tfrac{\epsilon_s}{1-e^{-\epsilon_s}})) + \sqrt{\tfrac{n\epsilon_s^2}{2} ln(\tfrac{1}{\delta})}\}.$$

We fix δ as $\frac{1}{|D|^{1.1}}$ to ensure that delta is less than $\frac{1}{|D|}$.

BR composition on the exponential mechanism outperforms all other composition techniques[28], and we adopt it to calculate $\epsilon_{partial}$ in our DP-selection process. An alternative method of combining differential privacy mechanisms is parallel composition, which can provide significantly better bounds but is only applicable in certain cases where the data can be partitioned. In this approach, the dataset is partitioned into non-overlapping subsets, and a differentially private mechanism is applied to each subset separately. Since each individual's data appears in only one subset, even if the dataset is partitioned into k subsets, each individual's data will only be subjected to one application of the mechanism. We mention formal definition of parallel composition in Lemma 7 and use it to investigate the privacy of our parallelized method.

Lemma 7. *If a differentially private mechanism $M(X)$ satisfies ϵ-differential privacy and we split a dataset into k disjoint chunks $(x_1, x_2, ..., x_k)$, then the mechanism which releases all of the results $(M(x_1), M(x_2), ..., M(x_k))$ satisfies ϵ-differential privacy.*

In the following section, we will examine the privacy guarantees of our approach.

4.4 Parallel DP-Select

DP-Select can be computationally expensive since the algorithm adds one data point at a time to the output dataset. To address this issue, we develop a parallelized version of

DP-Select that speeds up the algorithm by processing multiple data points simultaneously. Specifically, we parallelize the algorithm by partitioning the Pool and running multiple instances of DP-Select in parallel on each partition. We split the pool into sections to avoid repeated selection of data points. The resulting synthetic datasets generated from each partition are then combined to form the final synthetic dataset.

The parallelized version of DP-Select has the potential to be faster than the original algorithm since it can process multiple data points simultaneously, thereby reducing the overall computation time. In our experiments, we found that the parallelized version of DP-Select achieved a speedup of 10X on 32 CPU cores compared to the original algorithm.

In addition to parallelizing the algorithm to speed up running time, we can also naturally leverage parallel composition to improve the utility. One approach is to partition the original dataset into distinct sections and compute the marginals for each section separately. These marginals can then be used as different reference marginals for each parallel process in the selection process. In this case, we use parallel composition as defined in Lemma 7 to share the privacy budget among all processes. This approach ensures that each parallel process has the same privacy budget as the entire DP-Select algorithm, thus maintaining the overall privacy guarantee.

Using parallel composition increases the privacy budget for each repetition of the exponential mechanism, improving the utility of the output data. However, the size of each section should not be too small, as this would reduce the population size used to calculate marginal distributions, resulting in worse utility. Moreover, while parallel composition ensures that the marginal distribution of each section is close to a section of the original data, it does not guarantee that the marginal distribution of the union of the output of all sections is close to the original data. Therefore, the number of sections is a sensitive parameter in the algorithm, and we investigate its impact on utility in the results section.

It is important to note that parallel DP-Select can use any composition theorem to allocate the privacy budget to each parallel section, and parallel composition can also be used without a parallel algorithm. These are two independent concepts. Parallel DP-Select focuses on improving the run time of the algorithm, while parallel composition optimizes the utility of the output by balancing the privacy budget allocated to each repetition of the exponential mechanism with the amount of information captured in the marginals of each section. We demonstrate in Chapter 5 that parallel DP-Select using parallel composition is the most optimal version of DP-Select in terms of running time and utility performance.

4.5 Privacy guarantees

Algorithm 3: Privacy budget allocation

Input: ϵ_{total}: total privacy budget
$\epsilon\text{-}ratio$: ratio of $\epsilon_{dp\text{-}gan}$ to ϵ_{total}
p: number of parallel processes

; /* $\epsilon_{total} = \epsilon_{dp\text{-}gan} + \epsilon_{dp\text{-}select}$ */

$\epsilon_{dp\text{-}gan} \leftarrow \epsilon\text{-}ratio \times \epsilon_{total}$

$\epsilon_{dp\text{-}select} \leftarrow (1 - \epsilon\text{-}ratio) \times \epsilon_{total}$;

; /* replicate the $\epsilon_{dp\text{-}select}$ for each parallel process */

$\epsilon_{parallel} \leftarrow \epsilon_{dp\text{-}select}$;

; /* using BR composition */

; /* $\epsilon_{parallel} = min\{p\epsilon_s, p(\frac{\epsilon_s}{1-e^{-\epsilon_s}} - 1 - ln(\frac{\epsilon_s}{1-e^{-\epsilon_s}})) + \sqrt{\frac{p\epsilon_s^2}{2} ln(\frac{1}{\delta})}\}$ */

apply binary search to find closest ϵ_{br} *which satisfies* :

$\epsilon_{parallel} = p(\frac{\epsilon_{br}}{1-e^{-\epsilon_{br}}} - 1 - ln(\frac{\epsilon_{br}}{1-e^{-\epsilon_{br}}})) + \sqrt{\frac{p\epsilon_{br}^2}{2} ln(\frac{1}{\delta})}\}$;

$\epsilon_s = \max(\epsilon_{br}, \frac{\epsilon_{parallel}}{p})$;

In this section, we examine the privacy guarantees of our algorithm by analyzing its individual components. Our algorithm consists of two main components: DP-GAN and DP-selection, each of which uses a portion of the total privacy budget. According to Lemma 2, DP-GAN provides $\epsilon_{dp\text{-}gan}$-DP on the output of the generator, independent of the number of generated data points, ensuring that our data pool offers $\epsilon_{dp\text{-}gan}$-DP guarantees. Furthermore, according to Lemma 1, the post-processing feature of differential privacy guarantees that any random or deterministic function applied to the data pool, such as selecting a subset of data or normalizing over the features, would not incur additional privacy costs.

However, since we are directly using information from the marginal distributions of the original dataset to select the best candidate data points in repetitive tasks, we need to introduce a mechanism to ensure that the selection process does not reveal any information about the original dataset beyond the level of differential privacy that has been set. To achieve this, we employ the exponential mechanism, as stated in Lemma 4. Specifically, if we choose the data point with the best score based on the probabilities outlined in Lemma 4, we can guarantee $\epsilon_{partial}$-DP for a score function with sensitivity Δ_S. The challenge arises when we need to repeat this process $|D|$ times to obtain a full dataset with the desired size. In this case, we use the bounded range composition theorem, as stated in

Lemma 6, to determine the privacy budget allocation for each repetition, to ensure that the total privacy budget for the entire DP-selection process does not exceed $\epsilon_{dp\text{-}select}$.

So, as stated in Lemma 5, by applying the sequential composition of differential privacy, our entire algorithm satisfies ϵ_{total}-DP, where ϵ_{total} equals the sum of $\epsilon_{dp\text{-}gan}$ and $\epsilon_{dp\text{-}select}$. The privacy budget allocation for each repetition of the exponential mechanism in DP-selection is determined by the bounded range composition theorem, ensuring that the total privacy budget for the entire DP-selection process does not exceed $\epsilon_{dp\text{-}select}$.

As the output size $|D|$ increases, the privacy budget for each exponential mechanism repetition decreases, and the running time of DP-selection also increases. To address this, we use parallel DP-Select with parallel composition explained in 4.4. According to Lemma 7, if each parallel section satisfies $\epsilon_{parallel}$-DP, the entire DP-selection process satisfies $\epsilon_{dp\text{-}select}$-DP, where $\epsilon_{dp\text{-}select}$ equals $\epsilon_{parallel}$.

By using parallel composition, the privacy budget for each repetition of the exponential mechanism increases as the size of the output of each section decreases to $\frac{|D|}{\#sections}$, and parallel DP-Select allows faster running time. As previously mentioned, if the size of each section is too small, it can have a detrimental effect on the quality of the output data since it reduces the population size used to calculate marginal distributions, leading to a less accurate prediction of the utility. Furthermore, it is important to note that although parallelization ensures that the marginal distribution of each section is similar to a section of the original data, it does not guarantee that the marginal distribution of the output from all sections combined will be similar to that of the original data.

Lemma 8. *DP-Select is $(\epsilon_{total}, \delta)$-differentially private. The privacy budget of the algorithm is given by the following formula:*

$$\epsilon_{total} = \epsilon_{dp\text{-}gan} + min\{n\epsilon_s, n(\frac{\epsilon_s}{1-e^{-\epsilon_s}} - 1 - ln(\frac{\epsilon_s}{1-e^{-\epsilon_s}})) + \sqrt{\frac{n\epsilon_s^2}{2} ln(\frac{1}{\delta})}\} \tag{4.4}$$

where n represents the size of the output dataset for each parallel process, which is equal to $\frac{|D|}{\#sections}$. ϵ_s represents the privacy budget used for each exponential mechanism, and δ is set to $\frac{1}{|D|^{1.1}}$.

In this section, we have evaluated the privacy guarantees of our algorithm by analyzing the privacy properties of its individual components, DP-GAN and DP-Select, as well as the parallelized version of our algorithm, Parallel DP-Select. Algorithm 3 describes how the privacy budget is allocated across the different components of our algorithm. We have shown that our algorithm provides robust privacy guarantees while also generating high-quality synthetic data that can be effectively used for downstream tasks.

Chapter 5

Experiments and Results

In this section, we present the experimental results of our proposed algorithm for generating synthetic data with differential privacy guarantees. We begin by describing the experimental setup, including the division of the original dataset and the implementation and execution of the experiments. Then, we introduce the datasets used in our experiments, along with their characteristics. We also introduce the parameters of DP-Select that affect the utility, such as the privacy budget and the pool size.

Next, we explain our evaluation metrics, which include utility measures used by papers in this field, privacy guarantees, and running time. We evaluate the performance of our model and compare it to other state-of-the-art methods, such as DPGAN. We also analyze the effect of different parameters on the performance of our model and provide insights into the strengths and limitations of our approach. Our main questions to answer in this section are: how does our approach improve the utility of the synthesized dataset in comparison to a DP-GAN, and how do parameters like epsilon ratio and section number affect the performance?

It is important to note that when we refer to the classification accuracy of DP-GAN output in this section, we are referring to an independent DP-GAN model that uses the entire privacy budget (ϵ_{total}). This is different from the DP-GAN component used within our DP-Select algorithm, which only uses a portion of the privacy budget ($\epsilon_{dp\text{-}gan} = \epsilon\text{-}ratio \times \epsilon_{total} < \epsilon_{total}$).

5.1 Experiment setup

We implemented our data synthesis algorithm, as shown in Figure 4.1, using PyTorch [42]. To evaluate the performance of our framework, we follow the existing works for relational data synthesis and split the dataset into a training set $\mathcal{D}_{train}$, a validation set $\mathcal{D}_{valid}$, and a test set $\mathcal{D}_{test}$, with a ratio of 4:1:1, respectively. Prior to training, we performed preprocessing on the original dataset to ensure that all attributes were in numerical or one-hot-encoded form. We then trained a DP-GAN on the training set $\mathcal{D}_{train}$ to obtain optimized parameters for the discriminator and generator. We use the code provided by the authors of Daisy [22] to implement our DP-GAN baseline. This code is publicly available on GitHub [1]. During the training of DP-GAN, we take a snapshot of the model at the end of each of the ten epochs and evaluate the model on the validation set $\mathcal{D}_{valid}$ for each epoch. At the end of the training, we select the DP-GAN model snapshot with the best performance on the validation set to generate a synthetic pool. Subsequently, we applied the DP-select mechanism to the pool to obtain the synthetic output dataset $\hat{\mathcal{D}}$.

After obtaining $\hat{\mathcal{D}}$, we compared it with the original dataset $\mathcal{D}_{train}$ and the output of DPGAN with the same privacy budget as our entire process on both data utility and privacy protection. To ensure the robustness of our results, we repeated each experiment three times and we averaged the results over the three runs. In summary, our experimental setup consists of dataset splitting, DP-GAN model training, pool generation, DP-select, and result comparison. The performance of our model was evaluated using a variety of metrics, which will be discussed in section 5.1.3. In the following section, we will introduce the datasets used in our experiments.

5.1.1 Datasets

Our proposed method is tested on two tabular datasets: the Adult income dataset and the Forest CoverType dataset [15].

The Adult dataset [43] is in the social domain and contains information about an individual's annual income, which is influenced by various factors such as education level, age, gender, and occupation. The dataset includes 41,292 records with 6 numerical and 8 categorical attributes, and the targeted feature is income, which has two unique values: $> 50k$ and $\leq 50k$. Figure 5.1 shows a sample table from the Adult dataset, along with some synthesized samples.

[1] https://github.com/ruclty/Daisy

age	workclass	fnlwgt	education	education-num	marital-status	occupation	relationship	race	sex	capital-gain	capital-loss	hours-per-week	native-country	income
34	Private	112212	HS-grad	9	Married-civ-spouse	Adm-clerical	Husband	White	Male	0	1485	40	United-States	<=50K
30	Private	220148	HS-grad	9	Married-civ-spouse	Craft-repair	Husband	White	Male	0	1848	50	United-States	>50K
63	Self-emp-inc	38472	Some-college	10	Widowed	Sales	Not-in-family	White	Female	14084	0	60	United-States	>50K
35	Private	35945	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband	White	Male	0	0	40	United-States	>50K
26	Private	171114	Some-college	10	Never-married	Farming-fishing	Not-in-family	White	Female	0	0	38	United-States	<=50K
19	Private	278304	Some-college	10	Never-married	Other-service	Own-child	White	Female	0	0	15	United-States	<=50K
60	Private	186000	10th	6	Separated	Other-service	Not-in-family	White	Female	0	0	40	United-States	<=50K
32	Local-gov	255004	Some-college	10	Never-married	Craft-repair	Not-in-family	White	Male	0	0	40	United-States	<=50K
44	Private	165815	9th	5	Never-married	Machine-op-inspct	Not-in-family	White	Male	0	0	40	United-States	<=50K

age	workclass	fnlwgt	education	education-num	marital-status	occupation	relationship	race	sex	capital-gain	capital-loss	hours-per-week	native-country	income
34	Private	141009	HS-grad	9	Married-civ-spouse	Exec-managerial	Husband	White	Male	0	1365	40	United-States	<=50K
32	Private	93350	HS-grad	8	Married-civ-spouse	Craft-repair	Husband	White	Male	0	1655	39	Mexico	<=50K
32	Private	159909	Some-college	8	Married-civ-spouse	Exec-managerial	Husband	White	Male	0	1500	39	United-States	<=50K
37	Private	152437	Masters	9	Married-civ-spouse	Sales	Husband	White	Male	0	0	39	United-States	<=50K
33	Private	161588	HS-grad	8	Married-civ-spouse	Prof-specialty	Husband	White	Male	0	1506	40	United-States	>50K
55	Self-emp-not-inc	420419	Assoc-acdm	11	Separated	Other-service	Unmarried	Asian-Pac-Islander	Female	15545	0	71	Mexico	<=50K
39	Private	85696	Some-college	9	Married-civ-spouse	Sales	Husband	White	Male	0	0	39	Iran	<=50K
33	Private	156581	Some-college	9	Married-civ-spouse	Prof-specialty	Husband	White	Male	0	1490	40	United-States	>50K
33	Private	141251	HS-grad	9	Married-civ-spouse	Exec-managerial	Husband	White	Male	0	1377	40	United-States	<=50K

Figure 5.1: (a) the above table shows first ten rows from the original Adult dataset, and (b) the below table shows first ten rows from the synthetic dataset generated by our DP-Select algorithm with $\epsilon = 2$.

The CoverType dataset [44] contains tree observations from four areas of the Roosevelt National Forest in Colorado. The dataset includes information on tree type, shadow coverage, distance to nearby landmarks, soil type, and local topography. The simplified version of the CoverType dataset that we use has 116,204 records with 10 numerical and 2 categorical attributes, and the targeted feature is coverType, which has 7 unique values.

5.1.2 Parameters

In our analysis, we vary several parameters to evaluate the performance of our approach. One important parameter is the total privacy budget, which is the sum of the privacy budgets used in the DP-GAN and DP-selection sections ($\epsilon_{total} = \epsilon_{dp\text{-}gan} + \epsilon_{dp\text{-}select}$). This parameter is particularly important when comparing our algorithm to others and assessing the trade-off between privacy and utility.

The second parameter, the ϵ-*ratio*, represents the ratio between the privacy budget used in the DP-GAN section ($\epsilon_{dp\text{-}gan}$) and the total privacy budget used in the whole process ($\epsilon_{total} = \epsilon_{dp\text{-}gan} + \epsilon_{dp\text{-}select}$). A smaller ϵ-*ratio* means a larger part of the privacy budget was allocated to the DP-Select process instead of the DP-GAN. This parameter is particularly important for finding the best working ϵ-*ratio* and studying its impact on the output utility.

Another parameter is the distance metric used in our score function, which measures the similarity between the DP-GAN output and the original data. We explore the influence of this parameter on the output in our experiments.

We also use the pool size as a parameter to control the size of the DP-GAN output, which affects the performance of our approach. For example, a pool size equal to 10 means that the size of the DP-GAN output was 10 times the size of our desired output dataset.

Finally, we apply a parallelization technique to make our DP-selection algorithm faster by partitioning the data into several sections. The number of sections (parallel processes) used in this algorithm is a crucial parameter in studying the impact of parallelization on the performance of our approach.

We use these parameters throughout the remainder of this section to compare our method with DP-GAN and evaluate its performance. Additionally, we investigate the impact of these parameters on the algorithm and explain the rationale behind their usage.

5.1.3 Evaluation metrics

DP-Select is evaluated in three aspects: utility, privacy, and running time.

Utility evaluation. To evaluate the utility of our DP-select algorithm, we train a decision tree classifier with a depth of 10 on both the output dataset of DP-GAN and the output dataset of DP-select and measure their accuracies on a test dataset. We define the accuracy difference ($acc\text{-}diff$) between the two classifiers as:

$$acc\text{-}diff = acc_{dp\text{-}select} - acc_{dp}, \tag{5.1}$$

where $acc_{dp\text{-}select}$ and $acc_{dp\text{-}GAN}$ are the accuracies of the classifiers trained on the output datasets of DP-select and DP-GAN, respectively. By training the classifier on the output and testing it on a separate dataset, we can assess how well the synthetic data mimics the real data in terms of its ability to be used for downstream tasks such as classification.

Privacy Guarantee The DP-Select algorithm is designed to ensure differential privacy, which guarantees that the risk of identifying an individual in the original dataset from the generated synthetic data is limited. The privacy of the DP-Select algorithm is quantified by the total privacy budget allocated to the algorithm (ϵ_{total} including the privacy budget of DP-GAN $\epsilon_{dp\text{-}gan}$ and the privacy budget of DP selection $\epsilon_{dp\text{-}select}$), which limits the amount of information that can be leaked from the original dataset. Theoretical analyses have shown that the DP-GAN algorithm satisfies differential privacy with a privacy budget equal to $\epsilon_{dp\text{-}gan}$. And we show in Subsection 4.5 that DP selection process also satisfies $\epsilon_{dp\text{-}select}$-DP so our whole algorithm satisfies ϵ_{total}-DP. Therefore, there is no need to assess the privacy risk of the synthetic data generated by the DP-GAN using additional metrics.

Running time To evaluate the running time performance of our DP-Select algorithm, we measured its running time on a machine with 32 CPU cores. We also conducted experiments to investigate the effect of parallelization and the number of sections on the algorithm's running time. Specifically, we varied the number of sections from 1 to 10 and measured the running time for each setting. Our experiments show that the use of parallelization in DP-Select significantly reduces the running time compared to the non-parallelized version.

5.2 Results

In this section, we compare the performance of our differentially private data synthesis algorithm to DP-GAN for different privacy budgets (ϵ_{total}). Specifically, we evaluate the utility of the generated datasets in terms of accuracy and privacy guarantees for different values of ϵ_{total}. Additionally, we investigate the impact of the parameters introduced in the previous section on the performance of our method.

5.2.1 Comparison to DP-GAN

We conducted a comparison between our DP-Select algorithm and DP-GAN using different privacy budgets (ϵ_{total}), measured by the utility function of classification accuracy. The results, shown in Figure 5.2, demonstrate that our DP-Select algorithm outperforms DP-GAN in terms of utility for all privacy budgets considered. For example, when $\epsilon_{total} = 2$, DP-Select achieved a classification accuracy of 73.66% for the Adult dataset, while DP-GAN achieved only 70.7%. Similarly, when $\epsilon_{total} = 8$, DP-Select achieved a classification accuracy of 74.45% for the Adult dataset, while DP-GAN achieved only 73.55%. These results indicate that our algorithm can generate synthetic data with higher utility compared to DP-GAN.

Moreover, the performance improvement of DP-Select over DP-GAN was larger when ϵ_{total} was smaller. This can be attributed to the fact that DP-GAN may not be able to generate good results with high probability in a high privacy regime. However, by generating a larger pool of data with an even smaller ϵ (i.e., $\epsilon_{dp\text{-}gan}$) and selecting the data points that have closer marginals to the original data using DP-Select, we can improve the utility of the generated data. The points displayed in the chart were chosen using Pareto front selection among the average results of multiple runs with different epsilon values. Pareto front selection refers to a set of non-dominated solutions that are considered optimal if no objective can be improved without sacrificing at least one other objective.

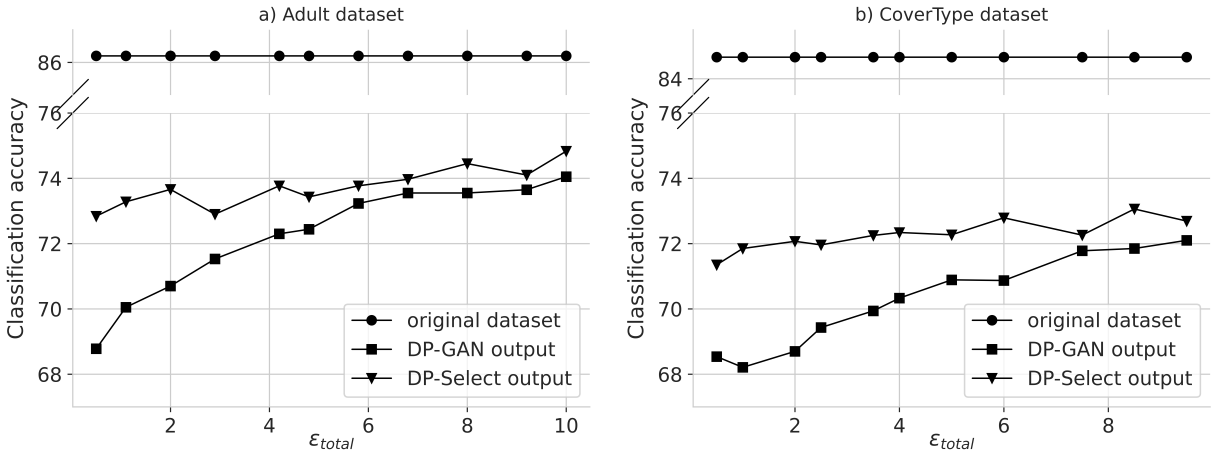

Figure 5.2: Comparing the utility of synthesized data of DP-Select with DP-GAN.

We choose the best achievable classification accuracy by DP-Select algorithm for a fixed value of ϵ_{total} by varying the other parameters: ϵ-*ratio*, pool size, distance metric, and number of sections. This was done to showcase the best performance achievable by DP-Select algorithm under different privacy budgets.

To evaluate our DP-Select algorithm in the remaining sections, we use the difference between the classification accuracy of DP-Select and DP-GAN, denoted as $acc\text{-}diff$.

5.2.2 Effect of epsilon ratio

In this section, we investigate the effect of the ϵ-*ratio* on the performance of DP-Select. To this end, we present the results of our experiments in the form of a bar chart, where we compare the classification accuracy difference ($acc\text{-}diff$) of DP-Select and DP-GAN for different values of the ϵ-*ratio*.

As shown in Figure 5.3, the performance of DP-Select is generally better when using smaller ϵ-*ratio*s. Specifically, when ϵ_{total} is 2, the smallest ϵ-*ratio* of 0.2 results in the largest accuracy difference of 2.96 in favor of DP-Select over DP-GAN. As the epsilon-ratio increases, the accuracy difference decreases, with the largest ϵ-*ratio* of 0.8 resulting in no improvement over DP-GAN.

A possible explanation for this trend is that smaller ϵ-*ratio* allows for more privacy budget to be allocated to DP-Select, resulting in a better selection of synthetic data points that have closer marginals to the original data. On the other hand, larger ϵ-*ratio* may lead

to a greater emphasis on the generation of synthetic data by DP-GAN, which may not always produce data that is representative of the original dataset.

Overall, our experiments suggest that the choice of the ϵ-*ratio* is an important consideration when using DP-Select, and that smaller ϵ-*ratio* may lead to improved performance.

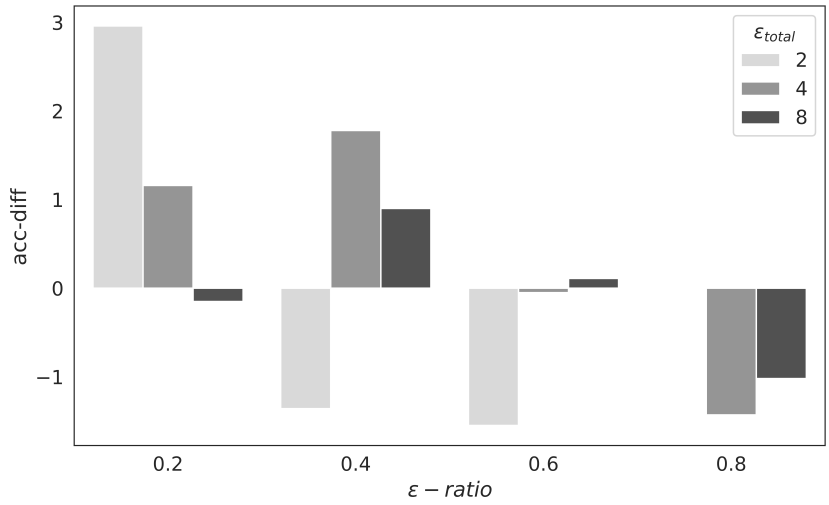

Figure 5.3: Effect of ϵ-*ratio* on performance of DP-Select

5.2.3 Effect of pool size

Another factor that can impact the performance of DP-Select is the size of the potential data point pool. To investigate this, we evaluated the accuracy difference ($acc\text{-}diff$) between DP-Select and DP-GAN for different pool sizes, while keeping the privacy budget (ϵ_{total}) constant.

As seen in the Figure 5.4, there is a general trend of increasing performance for DP-Select with larger pool sizes. For example, when ϵ_{total} is 2, increasing the pool size from 5 to 20 results in an $acc\text{-}diff$ improvement from 1.13 to 2.96. Similarly, when ϵ_{total} is 4, increasing the pool size from 5 to 15 results in an improvement from -0.72 to 1.38.

This trend can be explained by the fact that a larger pool of data points provides more diversity and potentially better coverage of the underlying data distribution. Therefore, DP-Select has a higher chance of selecting data points with closer marginals to the original data, leading to higher accuracy.

In summary, our experiments show that increasing the potential data point pool can improve the performance of DP-Select. However, this also comes at the cost of increased running time.

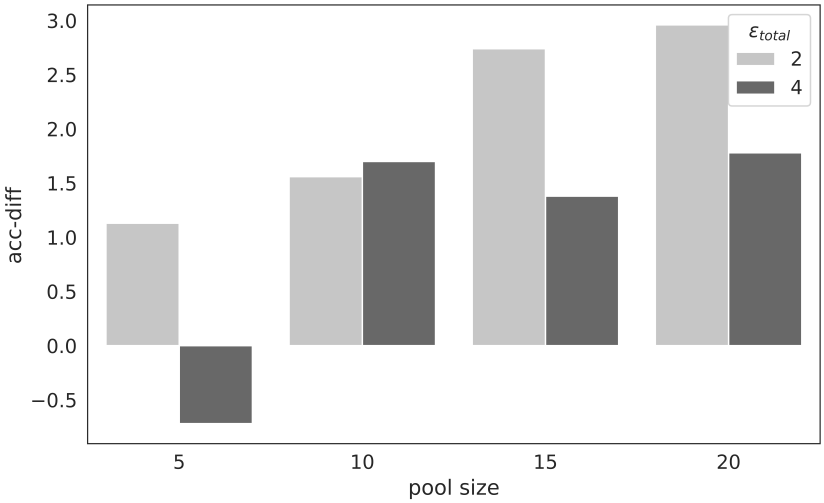

Figure 5.4: Effect of the pool size on performance of DP-Select

5.2.4 Effect of distribution distance metric

In this section, we evaluate the effect of different distance measures used in the score function of DP-Select on its performance. Specifically, we consider the Kolmogorov distance, the L2 density distance, and the density distance area. For each distance measure, we calculate the difference in classification accuracy between DP-Select and DP-GAN for different privacy budgets.

As seen in the Figure 5.5, the density distance metric yields the highest utility among the three distance measures considered. This observation suggests that the density distance metric is more suitable for measuring the similarity between the marginal distributions of the original and generated data points. One possible explanation is that the density distance area is a more flexible and fine-grained measure of distribution distance, as it takes into account the shape and location of the distributions. In contrast, the Kolmogorov distance only captures the maximum difference between the CDFs, which may not provide a complete picture of the differences in the distributions.

For our purpose of synthesizing data with similar marginals to the original data, measuring the distance between the PDFs seems more appropriate. This is because we want to ensure that the synthesized data has a similar probability distribution as the original data, which can be better captured by comparing the PDFs rather than the CDFs. L2 density distance also compares the PDFs, however, in this case, L1 distance might be preferred over L2 distance because it is more sensitive to differences between values at the tails of the distribution.

Overall, our results highlight the importance of choosing an appropriate distance measure in the score function of DP-Select for generating high-quality synthetic data.

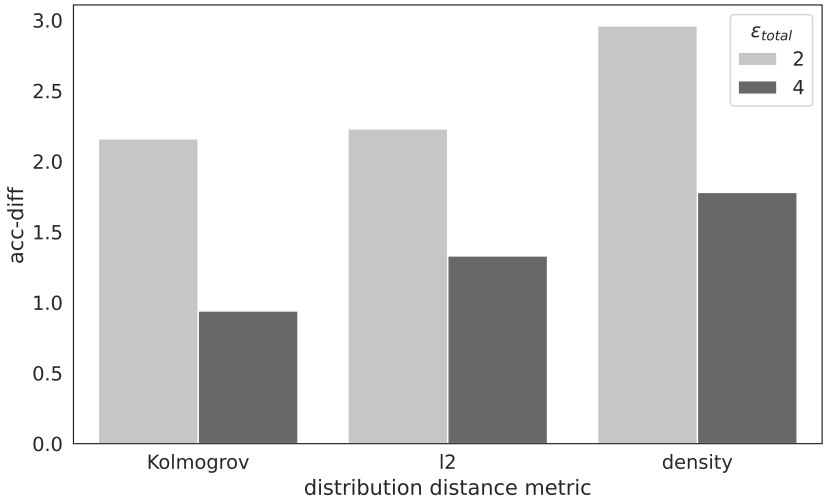

Figure 5.5: Effect of distribution distance metric on performance of DP-Select

5.2.5 Effect of number of sections

To investigate the effect of the number of sections on the performance of DP-Select, we conducted experiments using different numbers of sections ranging from 1 to 10. We computed the *acc-diff* for each number of sections and for different values of the total privacy budget.

As we can see from Figure 5.6, the output utility of DP-Select is highly dependent on the number of sections used in the computation. When the number of sections is too

small or too large, the utility performance of DP-Select is decreased, however the optimum choice (number of sections equal to 5) results in maximum utility.

The choice of the number of sections in the parallelized version of DP-select involves a trade-off between the partial privacy budget for each differentially private data selection iteration and the accuracy of the marginals of each section. A small number of sections allows for larger sections of the original data to be used to extract the marginal distributions, resulting in more accurate marginals. However, this means that the partial privacy budget for each iteration of differentially private data selection using the exponential mechanism is small. On the other hand, a larger number of sections means that we have a smaller part of the original dataset to get the marginals from, leading to less accurate marginals for the entire dataset. However, this provides a larger privacy budget in each iteration of the data selection process using the exponential mechanism. This means that we can pick the best data point candidate with a higher probability, even though the best candidate may not be the actual best candidate, as we are comparing it to a less accurate marginal distribution. Empirical results suggest that the optimal number of sections tends to be around 5.

To complement our analysis of the effect of the number of sections on the output utility of DP-Select, we also conducted experiments to evaluate the running time performance of the algorithm for different values of the number of sections. We used a fixed dataset size and varied the number of sections from 1 to 10, measuring the running time of each experiment.

As shown in Figure 5.7, we observed a decrease in the running time of DP-Select as the number of sections increased. This is because using more sections allows us to parallelize the computation and distribute the workload across multiple processes, reducing the total running time.

Overall, our experiments suggest that the optimal number of sections for DP-Select is around 5, as it provides the best utility and also results in a significant decrease in running time.

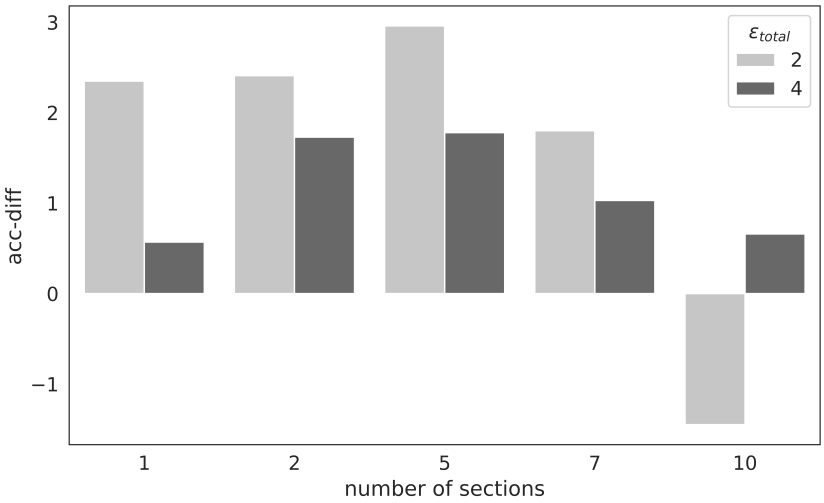

Figure 5.6: Effect of number of the sections on the performance of DP-Select

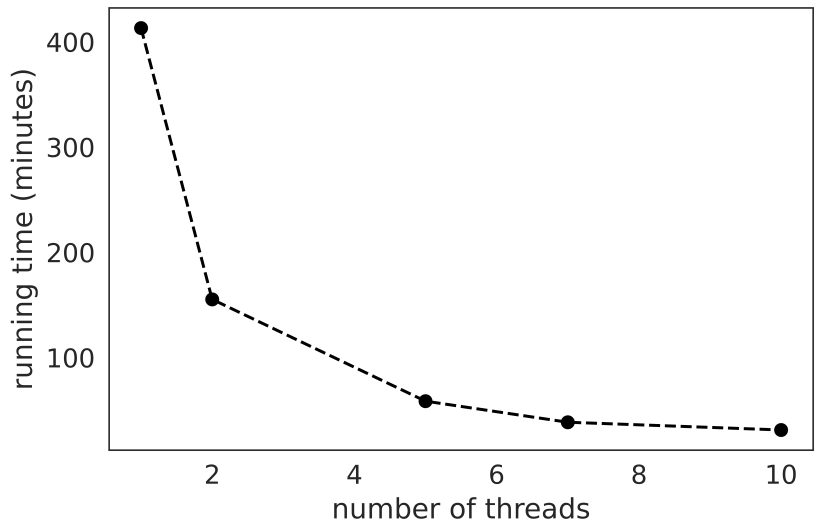

Figure 5.7: Effect of number of the sections on running time of DP-Select

Chapter 6

Conclusion

In this thesis, we proposed a novel method for improving the performance of DP-GANs on tabular data synthesis tasks. Our approach, called DP-Select, post-processes the output of DP-GAN by selecting data points that are more likely to be from the original dataset based on their marginal distributions.

Our experimental results demonstrate that DP-Select significantly improves the utility of synthesized data compared to DP-GAN alone, as measured by the classification accuracy metric. In particular, we show that DP-Select can achieve higher levels of utility while maintaining strong privacy guarantees, making it a promising solution for privacy-preserving data synthesis in terms of differential privacy. Furthermore, our experiments showed that DP-Select performs better for smaller privacy budgets, making it an attractive option for scenarios where the privacy budget is limited.

Moving forward, some open problems include comparing DP-Select to non-GAN data synthesis techniques, which may provide insights into the strengths and weaknesses of different methods. Additionally, exploring how we can modify the score function of DP-Select to serve other downstream tasks could lead to further improvements in the quality of synthesized data.

References

[1] Martin Abadi, Andy Chu, Ian Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. ACM, oct 2016.

[2] Martin Abadi, Andy Chu, Ian Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. ACM, oct 2016.

[3] John M. Abowd. The u.s. census bureau adopts differential privacy. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery Data Mining*, KDD '18, page 2867, New York, NY, USA, 2018. Association for Computing Machinery.

[4] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan, 2017.

[5] Sean Augenstein, H. Brendan McMahan, Daniel Ramage, Swaroop Ramaswamy, Peter Kairouz, Mingqing Chen, Rajiv Mathews, and Blaise Aguera y Arcas. Generative models for effective ml on private, decentralized datasets, 2020.

[6] Mrinal Baowaly, Chia-Ching Lin, Chao-Lin Liu, and Kuan-Ta Chen. Synthesizing electronic health records using improved generative adversarial networks. *Journal of the American Medical Informatics Association*, 26:228–241, 04 2019.

[7] Vincent Bindschaedler, Reza Shokri, and Carl A. Gunter. Plausible deniability for privacy-preserving data synthesis, 2017.

[8] Yang Cao and Linda Petzold. Accuracy limitations and the measurement of errors in the stochastic simulation of chemically reacting systems. *Journal of Computational Physics*, 212(1):6–24, 2006.

[9] Haipeng Chen, Sushil Jajodia, Jing Liu, Noseong Park, Vadim Sokolov, and V. S. Subrahmanian. Faketables: Using gans to generate functional dependency preserving tables with bounded real data. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 2074–2080. International Joint Conferences on Artificial Intelligence Organization, 7 2019.

[10] Qingrong Chen, Chong Xiang, Minhui Xue, Bo Li, Nikita Borisov, Dali Kaarfar, and Haojin Zhu. Differentially private data generative models, 2018.

[11] Rui Chen, Qian Xiao, Yu Zhang, and Jianliang Xu. Differentially private high-dimensional data publication via sampling-based inference. KDD '15, page 129–138, New York, NY, USA, 2015. Association for Computing Machinery.

[12] Edward Choi, Siddharth Biswal, Bradley Malin, Jon Duke, Walter F. Stewart, and Jimeng Sun. Generating multi-label discrete patient records using generative adversarial networks, 2018.

[13] Bolin Ding, Janardhan Kulkarni, and Sergey Yekhanin. Collecting telemetry data privately, 2017.

[14] Jinshuo Dong, David Durfee, and Ryan Rogers. Optimal differential privacy composition for exponential mechanisms and the cost of adaptivity, 2020.

[15] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017.

[16] David Durfee and Ryan Rogers. Practical differentially private top-k selection with pay-what-you-get composition, 2019.

[17] Cynthia Dwork. Differential privacy: A survey of results. In *Theory and Applications of Models of Computation: 5th International Conference, TAMC 2008, Xi'an, China, April 25-29, 2008. Proceedings 5*, pages 1–19. Springer, 2008.

[18] Cynthia Dwork. *Differential Privacy*, pages 338–340. Springer US, Boston, MA, 2011.

[19] Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. Our data, ourselves: Privacy via distributed noise generation. In Serge Vaudenay, editor, *Advances in Cryptology - EUROCRYPT 2006*, pages 486–503, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[20] Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.*, 9(3–4):211–407, aug 2014.

[21] Cynthia Dwork, Guy N. Rothblum, and Salil Vadhan. Boosting and differential privacy. In *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, pages 51–60, 2010.

[22] Ju Fan, Tongyu Liu, Guoliang Li, Junyou Chen, Yuwei Shen, and Xiaoyong Du. Relational data synthesis using generative adversarial networks: A design space exploration. *CoRR*, abs/2008.12763, 2020.

[23] Liyue Fan. A survey of differentially private generative adversarial networks. 2020.

[24] Dhami D.S. Kersting K. Fang, M.L. Dp-ctgan: Differentially private medical data generation using ctgans, 2022.

[25] Marco Gaboardi, Emilio Jesús Gallego Arias, Justin Hsu, Aaron Roth, and Zhiwei Steven Wu. Dual query: Practical private query release for high dimensional data, 2015.

[26] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014.

[27] Moritz Hardt, Katrina Ligett, and Frank McSherry. A simple and practical algorithm for differentially private data release, 2012.

[28] Thomas Humphries. Differentially private simple genetic algorithms, Nov 2021.

[29] Abou-El-Ela Hussien, Nermin Hamza, and Hesham Hefny. Attacks on anonymization-based privacy-preserving: A survey for data mining and data publishing. *Journal of Information Security*, 04:101–112, 01 2013.

[30] James Jordon, Lukasz Szpruch, Florimond Houssiau, Mirko Bottarelli, Giovanni Cherubin, Carsten Maple, Samuel N. Cohen, and Adrian Weller. Synthetic data – what, why and how?, 2022.

[31] Pei-Hsuan Lu, Pang-Chieh Wang, and Chia-Mu Yu. Empirical evaluation on synthetic data generation with generative adversarial network. pages 1–6, 06 2019.

[32] Josep Mateo-sanz, Francesc Sebe, and Josep Domingo-Ferrer. Outlier protection. 09 2004.

[33] Ryan McKenna, Gerome Miklau, and Daniel Sheldon. Winning the nist contest: A scalable and general approach to differentially private synthetic data, 2021.

[34] Ryan McKenna, Daniel Sheldon, and Gerome Miklau. Graphical-model based estimation and inference for differential privacy, 2019.

[35] Frank McSherry and Kunal Talwar. Mechanism design via differential privacy. pages 94–103, 11 2007.

[36] Ilya Mironov. Rényi differential privacy. In *2017 IEEE 30th Computer Security Foundations Symposium (CSF)*. IEEE, aug 2017.

[37] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets, 2014.

[38] Nicolas Papernot, Martín Abadi, Úlfar Erlingsson, Ian Goodfellow, and Kunal Talwar. Semi-supervised knowledge transfer for deep learning from private training data, 2017.

[39] Nicolas Papernot, Shuang Song, Ilya Mironov, Ananth Raghunathan, Kunal Talwar, and Úlfar Erlingsson. Scalable private learning with pate, 2018.

[40] Noseong Park, Mahmoud Mohammadi, Kshitij Gorde, Sushil Jajodia, Hongkyu Park, and Youngmin Kim. Data synthesis based on generative adversarial networks. *Proceedings of the VLDB Endowment*, 11(10):1071–1083, jun 2018.

[41] Yubin Park and Joydeep Ghosh. Pegs: Perturbed gibbs samplers that generate privacy-compliant synthetic data. *Transactions on Data Privacy*, 7:253–282, 12 2014.

[42] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[43] UCI Machine Learning Repository. Adult Data Set. https://archive.ics.uci.edu/ml/datasets/adult.

[44] UCI Machine Learning Repository. Covertype Data Set. http://archive.ics.uci.edu/ml/datasets/covertype.

[45] Ryan Rogers, Adrian Rivera Cardoso, Koray Mancuhan, Akash Kaura, Nikhil Gahlawat, Neha Jain, Paul Ko, and Parvez Ahammad. A members first approach to enabling linkedin's labor market insights at scale, 2020.

[46] Lucas Rosenblatt, Xiaoyan Liu, Samira Pouyanfar, Eduardo de Leon, Anuj Desai, and Joshua Allen. Differentially private synthetic data: Applied evaluations and enhancements, 2020.

[47] Reihaneh Torkzadehmahani, Peter Kairouz, and Benedict Paten. Dp-cgan: Differentially private synthetic data and label generation, 2020.

[48] Giuseppe Vietri, Grace Tian, Mark Bun, Thomas Steinke, and Zhiwei Steven Wu. New oracle-efficient algorithms for private synthetic data release, 2020.

[49] Liyang Xie, Kaixiang Lin, Shu Wang, Fei Wang, and Jiayu Zhou. Differentially private generative adversarial network, 2018.

[50] Lei Xu, Maria Skoularidou, Alfredo Cuesta-Infante, and Kalyan Veeramachaneni. Modeling tabular data using conditional gan, 2019.

[51] Lei Xu and Kalyan Veeramachaneni. Synthesizing tabular data using generative adversarial networks, 2018.

[52] Jinsung Yoon, James Jordon, and Mihaela van der Schaar. PATE-GAN: Generating synthetic data with differential privacy guarantees. In *International Conference on Learning Representations*, 2019.

[53] Jun Zhang, Graham Cormode, Cecilia M. Procopiuc, Divesh Srivastava, and Xiaokui Xiao. Privbayes: Private data release via bayesian networks. *ACM Trans. Database Syst.*, 42(4), oct 2017.

[54] Xinyang Zhang, Shouling Ji, and Ting Wang. Differentially private releasing via deep generative model (technical report), 2018.

[55] Zhikun Zhang, Tianhao Wang, Ninghui Li, Jean Honorio, Michael Backes, Shibo He, Jiming Chen, and Yang Zhang. *PrivSyn: Differentially Private Data Synthesis*. 2020.

[56] Zilong Zhao, Aditya Kunar, Hiek Van der Scheer, Robert Birke, and Lydia Y. Chen. Ctab-gan: Effective table data synthesizing, 2021.