DP-2Stage: Adapting Language Models as Differentially Private Tabular Data Generators

Anonymous authors
Paper under double-blind review

Abstract

Generating tabular data under differential privacy (DP) protection ensures theoretical privacy guarantees but poses challenges for training machine learning models, primarily due to the need to capture complex structures under noisy supervision signals. Recently, pre-trained Large Language Models (LLMs) – even those at the scale of GPT-2 – have demonstrated great potential in synthesizing tabular data. However, their applications under DP constraints remain largely unexplored. In this work, we address this gap by applying DP techniques to the generation of synthetic tabular data. Our findings shows that LLMs face difficulties in generating coherent text when fine-tuned with DP, as privacy budgets are inefficiently allocated to non-private elements like table structures. To overcome this, we propose DP-2Stage, a two-stage fine-tuning framework for differentially private tabular data generation. The first stage involves non-private fine-tuning on a pseudo dataset, followed by DP fine-tuning on a private dataset. Our empirical results show that this approach improves performance across various settings and metrics compared to directly fine-tuned LLMs in DP contexts. We release our code and setup at `https://anonymous.4open.science/r/DP-2Stage-B8B1/README.md`.

1 Introduction

Tabular data is one of the most prevalent data types, providing structured information in rows and columns, and has been extensively used across various applications. Due to privacy concerns, tabular data can not be directly shared. A widely adopted approach to address this issue is to train synthetic tabular data generators under differential privacy (DP) (Dwork, 2006). Different model classes have been proposed, from marginal-based statistical models (McKenna et al., 2022; Zhang et al., 2017) to prominent Generative Adversarial Networks (GANs)-based tabular data generators (Xu et al., 2019), trained with DP-Stochastic Gradient Descent (DP-SGD) (Abadi et al., 2016). These models aim to replicate the marginal distribution of the original data while preserving the utility of the synthetic data. At the same time, they enforce a strict theoretical upper bound on privacy leakage, enabling users to generate realistic data samples without compromising privacy. Despite these advancements, existing techniques continue to grapple with major challenges, such as scalability limitations and difficulties in accurately modeling marginal distributions. These difficulties can be traced back to the intricate nature of tabular data and the challenges associated with training under noisy DP conditions.

Recently, pre-trained Large Language Models (LLMs) (Radford et al., 2019; Touvron et al., 2023) have demonstrated remarkable adaptability to tasks they have never been specifically trained for (Wei et al., 2021; Chung et al., 2022; Wang & Fritz, 2024). Acting as compact knowledge bases, LLMs present a promising opportunity for developing practical tabular data generators by leveraging pre-existing knowledge – an ability not currently feasible with traditional tabular data generators.

Recent work from Borisov et al. (2023) showcased how LLMs can effectively be used for synthetic tabular data generation, by representing each cell in the format "`<key> is <value>,`". While LLMs hold promise as generative priors for tabular data, adapting them under DP constraints introduces unique challenges. In DP-SGD, noise is added to the gradients to ensure privacy, which affects all tokens, including key tokens that may not be privacy-sensitive, such as column names or structural markers (e.g., "`<key> is ,`"). This indiscriminate application of noise can disrupt the model's ability to maintain the structural integrity and semantic clarity required for tabular data generation. As a result, both the utility and fidelity of the generated synthetic data are negatively affected, highlighting the need for more targeted noise application techniques to minimize such impacts while adhering to DP constraints.

In this work, we demonstrate that directly fine-tuning LLMs under DP constraints leads to sub-optimal performance. Our findings, shown in Figure 3, reveal that LLMs struggle to generate coherent and structured text when fine-tuned with DP. This challenge stems from inefficient privacy budget allocation to potentially non-private elements, such as table structures or non-functional tokens.

To address this limitation, we propose **DP-2Stage**, a two-stage fine-tuning framework for tabular data generation. In the first stage, DP-2Stage fine-tunes the pre-trained LLM non-privately on a pseudo dataset, allowing the model to learn task-specific structures and patterns without consuming the privacy budget. In the second stage, fine-tuning proceeds on the private dataset with DP constraints (see Figure 1 for an overview of the approach). By learning the structural patterns in the first stage, the DP-constrained fine-tuning can focus on preserving the privacy of the data values.

For constructing pseudo datasets in the first stage, we investigated two strategies: (1) drawing data independently from a uniform distribution using statistics constructed from the private dataset (DP-2Stage-U) and (2) using an out-of-distribution public dataset that is unrelated to the private data (DP-2Stage-O). These approaches offer varying levels of privacy protection, with DP-2Stage-O providing stronger protection as it does not require prior knowledge of the private data.

We evaluated DP-2Stage on two datasets and observed improvements over standard DP fine-tuning across the metrics considered. Notably, DP-2Stage-U exhibited faster inference times (up to 21x) compared to DP-2Stage-O and DP-Standard. Additionally, we observe that fine-tuning with DP while avoiding column shuffling, as used in Borisov et al. (2023), yielded better performance under DP constraints but underperformed in Non-DP scenarios. Meanwhile, DP benchmarks considered in this work (e.g., DP-CTGAN) demonstrated competitive performance on utility-based metrics but underperformed in marginal-based metrics compared to our proposed DP-2Stage framework.

We summarize our contributions as follows:

- Proposed DP-2Stage, a two-stage fine-tuning framework for LLM-based tabular data generation under DP, which fine-tunes on pseudo datasets non-privately to learn task-specific structures, enabling more efficient use of the privacy budget during DP fine-tuning.
- Achieved 3-7% relative reductions in perplexity and 1-2% relative improvements in marginal-based metric (averaged over two datasets) compared to standard DP fine-tuning, with experiments repeated five times per model and four synthetic datasets generated per model. Additionally, DP-2Stage-U demonstrated significantly faster inference—up to 21x faster than DP-Standard and DP-2Stage-O.
- Despite the effectiveness of our approach, our findings highlight the complexity of fine-tuning LLMs under DP for tabular data generation, emphasizing the need for further research to advance privacy-preserving LLM-based tabular data generators. To support future advancements, we release our code and provide a detailed discussion to guide further investigation.

2 Related work

Tabular data generation. As a prominent solution for data sharing, a large and growing body of research has been focused on tabular data generation. These methods can be generally categorized into marginal-based and deep learning-based methods. Marginal-based methods, such as those developed by Zhang et al. (2017); Aviñó et al. (2018); McKenna et al. (2022), view each table column as a distinct random variable and model them using specific distributions. This approach often requires prior understanding or domain expertise in the data, hindering their scalability. Nevertheless, marginal-based methods (e.g. McKenna et al. (2022)) have

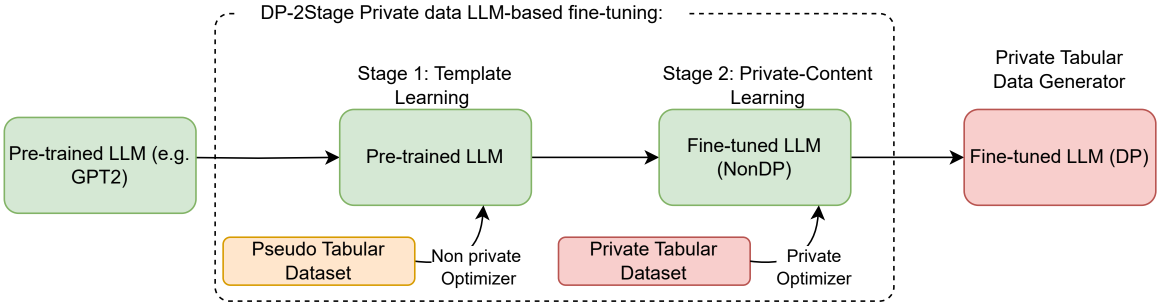
DP-2Stage Private data LLM-based fine-tuning:
Pre-trained LLM (e.g. GPT2)
Stage 1: Template Learning
Pre-trained LLM
Pseudo Tabular Dataset
Non private Optimizer
Stage 2: Private-Content Learning
Fine-tuned LLM (NonDP)
Private Tabular Dataset
Private Optimizer
Private Tabular Data Generator
Fine-tuned LLM (DP)

Figure 1: **Overview of DP-2Stage.**

proven effective in integrating DP guarantees, making them a compelling choice for privacy-preserving data generation. On the other hand, deep learning (DL)-based methods such as (Choi et al., 2017; Park et al., 2018; Xu et al., 2019; Kotelnikov et al., 2023) leverage cutting-edge generative models such as Variational Autoencoders (VAEs) (Kingma & Welling, 2014), Generative Adversarial Networks (GANs) (Goodfellow et al., 2014), or diffusion models (Sohl-Dickstein et al., 2015) to synthesize tabular data. These models can capture complex patterns and correlations present in tabular data, often producing high-fidelity outputs. However, by default, these prominent DL-based tabular data generators (e.g Xu et al. (2019); Kotelnikov et al. (2023)) do not adhere to DP standards, which is a significant limitation for applications requiring strong privacy guarantees. Recent advances have introduced privacy-preserving mechanisms, such as Differentially Private Stochastic Gradient Descent (DPSGD) (Abadi et al., 2016), to modify these models for DP compliance (Xie et al., 2018). Despite this progress, ensuring DP within tabular-based deep learning models poses unique challenges due to the complexity of the preprocessing steps. For example, proposed improvements like mode-specific normalization introduced in (Xu et al., 2019) must also satisfy DP, as preprocessing steps themselves can impact privacy (Ponomareva et al., 2023). In this work, we leverage pre-trained LLMs for tabular data generation, taking advantage of their natural language processing capabilities, and investigate their potential for generating tabular data that is compliant with DP standards.

Large Language Models (LLMs). Language models have been extensively studied over the years, evolving from statistical models (Jelinek, 1998) and recurrent neural networks (Hochreiter & Schmidhuber, 1997) to the latest transformer-based architectures (Vaswani et al., 2017). These advancements, supported by the attention mechanisms and rich text datasets, have led to the emergence of large-scale language models (Radford et al., 2019; Brown et al., 2020; Touvron et al., 2023; Achiam et al., 2023), a new generation of language models. These models, together with various fine-tuning strategies (Hu et al., 2021; Zhou et al., 2022), have enabled several interesting applications (Borisov et al., 2023; Wang & Fritz, 2024). Borisov et al. (2023) proposed using pre-trained LLMs for synthesizing tabular data in a non-private setting by converting tabular data into text-like formats, significantly improving the performance and paving a new path toward LLM-based tabular generators. In light of the potential, our work delves into applying these advanced models to a relatively unexplored domain: the generation of tabular data under differential privacy constraints. While previous studies (Yu et al., 2021; Tito et al., 2024) have explored private fine-tuning of LLM, our focus specifically targets tabular data generation, a unique challenge that diverges from the broader scope of text comprehension. Recently, a concurrent work (Tran & Xiong, 2024) shared a similar spirit of two-stage training. However, this work does not provide insight into the impact of pseudo data used in the first stage and examines a different family of LLM and fine-tuning strategy compared to this work. Our focus is on GPT-2, as utilized in the pioneering work by Borisov et al. (2023), to gain deeper insights into how effective design choices in non-differentially private (Non-DP) settings translate to DP contexts. We offer a detailed analysis of pseudo-data choice, the impact of column shuffling on generation of coherent synthetic tabular data, identify the limitations of the two-stage approach, and discuss open challenges.

Example table

age	sex	income
32	female	>50k

Iteration 1: age is 32, sex is female, income is >50k.
Iteration 2: income is >50k, age is 32, sex is female.
Iteration 3: sex is female, income is >50k, age is 32.

Figure 2: **Illustration of column shuffling.** The order of entries is permuted in each iteration.

3 Background

3.1 Language Models for Tabular Data

Language models are designed to model the probability of text sequences. Consider a text corpus, denoted as $\mathcal{S} = \{\boldsymbol{w}_i\}_{i=1}^{N}$, comprising N sentences. Each sentence $\boldsymbol{w} = (t_1, \dots, t_K)$ within $\mathcal{S}$ consists of an ordered sequence of K tokens. These tokens may represent either whole words or parts of words, generated through a tokenization process such as Byte-Pair Encoding (BPE) introduced by Sennrich et al. (2015). The tokenization process can be expressed as $(t_1, \dots, t_K) = \texttt{tokenizer}(\boldsymbol{w}_i)$. The probability of a given sentence $\boldsymbol{w}$ is formulated as:

$$p(\boldsymbol{w}_i) = p(t_1, \dots, t_K) = \prod_{k=1}^{K} p(t_k | t_{<k}), \tag{1}$$

where $t_{<k} = (t_1, \dots, t_{k-1})$ represents all tokens preceding the k-th token, and $p(t_k|t_{<k})$ denotes the conditional probability of token t_k given all prior tokens.

In the context of tabular data, data records $\boldsymbol{w} = (\mathcal{K}, \mathcal{V})$ are defined as a collection of key-value pairs, where $\mathcal{K} = \{k_q\}_{i=1}^{Q}$ represents the set of keys and $\mathcal{V} = \{v_m\}_{j=1}^{M}$ represents the corresponding set of values. To enable LLMs to process these records, we define the serialization process below, which converts the key-value pairs into a format understandable to models.

Definition 3.1 (Serialization) *Let f represent **template**, which defines the general pattern for organizing tabular data. Using GReaT serialization (Borisov et al., 2023), the template is expressed as f = "`<key> is <value>,`", specifying how keys ($k \in \mathcal{K}$), values ($v \in \mathcal{V}$), and non-functional elements ($c \in \mathcal{C}$, such as "is" and ",") combine to form a record. An input dataset denoted as $\mathcal{S}$, is a realized instance of this f, instantiated with tokens (k, v) from a record $\boldsymbol{w} = (\mathcal{K}, \mathcal{V})$. For example, with $k = \{$`age`$\}$ and $v = \{32\}$, $\mathcal{S}$ is instantiated as $f(k, v)$ = "`age is 32,`".*

Following tokenization, tokens associated with elements in the key or value set are denoted as $t_i \in \mathcal{K}$ or $t_j \in \mathcal{V}$, with i and j being the position indices, respectively. Note that, due to tokenization, the number of tokens for the keys $\mathcal{K}$ and the values $\mathcal{V}$ are not necessarily equal.

With definition 3.1, any content can be applied to a specified template.

Column Shuffling. Column shuffling involves randomly altering the order of column-aligned values within each batch during training. For example, given columns *age*, *sex*, and *income* with values *32*, *female*, and *>50k*. A detailed illustration is provided in Figure 2. This mechanism happens at every iteration and has been shown to effectively prevent the model from relying on spurious dependency in Non-DP settings. Disabling shuffling fixes the order across iterations, potentially leading to overfitting due to reduced variability. The primary purpose of shuffling is to enhance generalization by disrupting order-based patterns. However, it complicates DP training due to gradient perturbations, often resulting in higher perplexity and different behavior compared to Non-DP models, as shown in Figure 3.

3.2 Differential Privacy

Differential Privacy mechanisms enable the confidential disclosure of information about a dataset by perturbing a function of the input dataset. This ensures that any information capable of distinguishing a specific record from the remainder of the dataset is constrained, as outlined by Dwork et al. (2014). In this work, we consider privacy-preserving tabular data generators to ensure that any information leakage from the generated data is bounded by DP. We review the necessary definitions, the threat model, and the privacy model below.

Definition 3.2 *(Differential Privacy (Dwork et al., 2014)) A randomized mechanism* $\mathcal{M}$ *with range* $\mathcal{R}$ *satisfies* (ε, δ)*-differential privacy, if for any two adjacent datasets* E *and* E'*, i.e*$E' = E \cup \{x\}$ *for some* x *in the data domain (or vice versa), and for any subset of outputs* $O \subseteq \mathcal{R}$*, it holds that*

$$\Pr[\mathcal{M}(E) \in O] \leq e^{\varepsilon} \Pr[\mathcal{M}(E') \in O] + \delta \tag{2}$$

where ε is the privacy budget and δ is the probability of the mechanism failing.

Intuitively, this guarantees that an adversary, provided with the output $\mathcal{M}$, can draw almost the same conclusions (up to ε with probability larger than $1 - \delta$) about any record no matter if it is included in the input of $\mathcal{M}$ or not (Dwork et al., 2014). That is, for any record owner, a privacy breach is unlikely due to their participation in the dataset.

Definition 3.3 *(Gaussian Mechanism (Dwork et al., 2014)) Let* $f : \mathbb{R}^n \to \mathbb{R}^d$ *be an arbitrary function that maps* n*-dimensional input to* d *logits with sensitivity being:*

$$S = \max_{E,E'} \|f(E) - f(E')\|_2 \tag{3}$$

over all adjacent datasets E *and* $E' \in \mathcal{E}$*. The Gaussian Mechanism* $\mathcal{M}_\sigma$*, parameterized by* σ*, adds noise into the output i.e.,*

$$\mathcal{M}_\sigma(x) = f(x) + \mathcal{N}(0, \sigma^2 \mathbf{I}_n). \tag{4}$$

Threat Model. We outline a threat model where the goal of the adversary is to infer information about individuals in the training dataset by launching diverse privacy attacks. One such attack is the Membership Inference Attack (Shokri et al., 2017; Hilprecht et al., 2019; Chen et al., 2020), which determines whether a specific data point was included in the model's training set. We imagine a scenario involving a strong adversary who possesses complete access to the model post-training, known as "white-box access". This scenario is considerably more critical than a "black-box access" setting, where the adversary is limited to interacting with and analyzing the synthetic data produced by the model, without insight into its internal workings. In addition, the adversary is computationally unbounded. The impact of the threat model is the potential exposure of sensitive individual data from the training set, thereby compromising data privacy and undermining trust in the model.

Privacy Model. To defend against such threat, we aim to develop a solution that protects against potential privacy attacks targeting individuals in our training dataset. To achieve this, we employ Differential Privacy (DP), which was specifically designed to address this challenge. DP provides the assurance of plausible deniability, meaning that any potential privacy infringement on an individual – for example, a patient in the dataset – cannot be conclusively linked to their participation in the model's training phase up to ε with probability larger than $1 - \delta$. Consider a scenario where an insurance company decides to raise a patient's insurance premium after analyzing a model trained on a medical dataset. In such a case, owing to the principles of DP, the increase cannot be directly attributed to the inclusion of that patient's data in the training process. We provide further details in Appendix A.

4 DP-2Stage

While large language models (LLMs) demonstrate impressive capabilities in generating synthetic tabular data in non-differentially private (Non-DP) settings, as we will later show, fine-tuning them under DP constraints

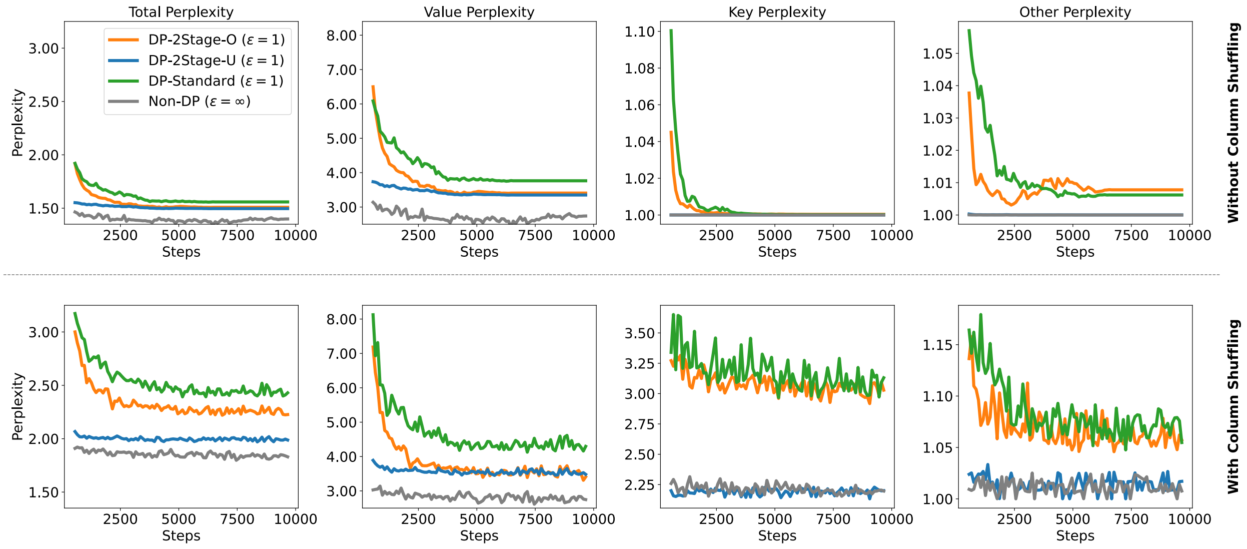

Figure 3: **DP-2Stage (Ours) vs. Standard DP fine-tuning on the Adult dataset with $\varepsilon = 1$.** DP-2Stage-O and DP-2Stage-U denotes using Out-of-distribution dataset (Airline) and data sampled independently from a Uniform distribution with statistics from the Adult dataset as a pseudo-dataset in the first stage. Perplexity results are displayed from left to right for all words, values, keys, and non-functional words (c.f. section 3.1). The top plot shows model trained without column shuffling, and the bottom shows model trained with column shuffling. Total and Value Perplexity for top and bottom plots have fixed y-axis for ease of comparison.

remains challenging for tabular data generation. As illustrated in Figure 3 (bottom), *standard* DP training, where the LLM model is directly fine-tuned using DPSGD (Abadi et al., 2016), often results in suboptimal performance in terms of the perplexity metric. We hypothesize that this is due to the inefficient allocation of the privacy budget during the learning of data structures. Models employing this approach struggle to reduce the perplexity of keys (e.g., column names) and other non-functional words, despite these elements typically posing minimal privacy risks. Consequently, applying DP to such components not only wastes the privacy budget but also diminishes the overall utility of the model.

Motivated by this observation, we propose DP-2Stage, a two-stage learning approach designed to progressively capture tabular data structures. An overview of our method is presented in Figure 1.

In the first stage (section 4.1), we train the model on non-private *pseudo* data synthesized based on (1) prior knowledge of the private data and (2) publicly available out-of-distribution dataset, guiding it to learn the underlying structure. In the second stage (section 4.2), the model is fine-tuned on the private data while ensuring DP protection. The goal of our two-stage approach is to decouple the process of learning the table's structure from learning the private content.

4.1 Stage 1: Template Learning

The goal of the first stage is to enable the model to learn the structure of the data table, such as identifying key-value relationships, without consuming any privacy budget. We achieve it by constructing a pseudo dataset that retain the structural information while eliminating privacy risks. To achieve this, we investigated two different methods for constructing a pseudo dataset $\widetilde{\mathcal{S}}$: (1) Sampled data from a uniform distribution using the private data statistics and (2) Out-of-distribution public data. Each method provides different levels of reliance on private data while aiming to preserve privacy and utility.

Independently Sampled data from a Uniform Distribution (uniform pseudo data). This dataset is designed to *resemble* the private dataset we aim to protect. We assume that column names are public information and can be utilized without any privacy concerns. Additionally, we assume that the data owner can provide the range (maximum and minimum values) for numerical columns and the list of categories for each categorical column. For each column, samples are *independently* drawn from a uniform distribution, where all elements have an equal probability of being selected using the statistics from the private data such as the range of numerical column, and the categories in each categorical column. This assumption is minimal compared to requiring detailed distributional information, which may be unrealistic. However, while this approach minimizes reliance on private data, it still presumes access to basic information, such as category labels and numerical ranges.

Out-of-distribution Public data (out-distribution pseudo data). In contrast to the previous method, this approach eliminates the reliance on private data by using a publicly available dataset. In this work, we refer to *Out-of-distribution* as any publicly available dataset that is not the private data. This approach avoids the need to infer statistical properties of the original dataset, relying instead on the structure provided by the public data. The model is trained directly on this public dataset, which serves as a *proxy* for the private data during the first stage. As we will show, despite not requiring any information from the original dataset, this method performs comparably to the uniformly sampled dataset in many cases and even outperforms it in certain scenarios.

Objective function. In the first stage, the model is trained using standard training protocols on the serialized version of the pseudo dataset. In particular, we fine-tune a pre-trained LLM p_θ, parameterized by θ, using cross-entropy loss for causal language modeling in a non-private setup. We formalize the loss function as follows.

$$\mathcal{L} = \mathbb{E}_{\boldsymbol{w} \in \widetilde{\mathcal{S}}} \left[- \sum_{t_k \in \boldsymbol{w}} \log p_\theta(t_k \mid t_{<k}) \right], \tag{5}$$

where $\widetilde{\mathcal{S}}$, $\boldsymbol{w}$, and t represent the pseudo dataset, individual sentences, and the corresponding tokens.

4.2 Stage 2: Private-Content learning

In the second stage, we introduce learning of the actual private content using DP-SGD (Abadi et al., 2016) on the private data $\mathcal{S} = (\mathcal{K}, \mathcal{C}, \mathcal{V})$. To encourage the model to prioritize learning value tokens, we apply a weighted loss that balances the learning between value tokens and non-private tokens (key and non-functional tokens). This approach builds on the insight that the stage 1 model has already effectively learned the non-private tokens (particularly when using uniform pseudo-data) and acquired the overall structure from the first stage based on the specified template. Since Stage 1 also learns the content (key and value) while learning the template, we assign a higher weight λ to value tokens $t \in \mathcal{V}$ in this second stage to reinforce the learning of private values as proposed by Tran & Xiong (2024). The complete loss function is defined as follows:

$$\mathcal{L} = \mathbb{E}_{(\mathcal{V},\mathcal{C},\mathcal{K}) \in \mathcal{S}} \left[-\lambda \sum_{t_j \in \mathcal{V}} \log p(t_j \mid t_{<j}) - (1-\lambda) \sum_{t_i \in \{\mathcal{K},\mathcal{C}\}} \log p(t_i \mid t_{<i}) \right], \tag{6}$$

where λ controls the emphasis on content learning.

The stage 2 model trained with out-distribution pseudo data is denoted as **DP-2Stage-O**, while the model trained with uniform pseudo data is denoted as **DP-2Stage-U**.

5 Experiments

In this section, we outline the experimental protocol and evaluate our proposed method against the DP-Standard method, other Non-LLM-based DP methods, and Non-DP approaches.

5.1 Implementation Details

GPT-2 Fine-tuning. We utilized the GPT-2 Radford et al. (2019) model from Hugging Face [1] and integrated Opacus [2] for differential privacy (DP) training, leveraging the BatchMemoryManager for efficient micro-batching. We *fully* fine-tuned the model with DP-Adam optimizer using a learning rate of $5e^{-5}$ and a linear scheduler, setting the maximum gradient norm $C = 1$ and a target delta $\delta = 1e^{-5}$. In the second stage of DP-2Stage, the parameter λ was consistently set to 0.65 across all experiments. To convert the table-to-text, we used the GReaT serialization method (Borisov et al., 2023). Unlike GReaT, we maintained a fixed column ordering for each sampled dataset for the DP approaches. The Non-DP benchmark using GPT-2, the DP-Standard model, and the stage 2 models of DP-2Stage are all trained for 10 epochs, while the stage 1 model of DP-2Stage is trained for 5 epochs.

GPT-2 Sampling. To sample from the model, we condition on the target token for each dataset. For example, in the Adult dataset, where "income" is the target token, sampling begins with the prompt 'income is ', and generation continues until all columns have been sampled. A rejection sampling approach was used to discard incomplete generations. However, for the approach with column shuffling enabled (Table 4), rejection sampling proved to be very slow and sometimes failed to complete any generation. In these cases, we employed an imputation-based approach, as detailed in Appendix B.1.

Baseline Models. We compare our approach to Non-LLM-based methods under DP and Non-DP settings. For Non-DP baselines, we considered CTGAN, TVAE (Xu et al., 2019) and VAE (adaptation of TVAE using standardized preprocessing technique) while DP baselines include DP-CTGAN, DP-GAN, and DP-VAE. For CTGAN and TVAE, we followed the implementation [3], while DP-CTGAN and DP-GAN were implemented using the SmartNoise SDK [4]. The numerical column preprocessing was conducted with $\varepsilon = 0.1$, while the remaining privacy budget was reserved for private training. For DP-VAE, we adapted the VAE model using SmartNoise preprocessing with $\varepsilon = 0.1$.

All experiments were conducted using NVIDIA A100-SXM4-40GB GPUs.

Datasets. We evaluated two tabular datasets: the Adult Income dataset, which contains over 30,000 samples and 15 attributes used to predict whether an individual's annual income exceeds \$50,000, hosted on the UCI Machine Learning Repository[5]; and the Airline Passenger Satisfaction dataset, which includes over 100,000 samples and 24 attributes related to passenger satisfaction, available on Kaggle[6]. Unlike the more commonly benchmarked Adult dataset, the Airline dataset offers a fresh perspective on the ability of deep learning models to capture complex relationships within diverse tabular data and large sample sizes. Table 1 summarizes the statistics of the tabular data. More details can be found in Appendix D.1.

	# Train	# Test	# N	# C	# Total
Adult	30932	16858	6	9	15
Airline	103904	24976	19	5	24

Table 1: **Tabular Dataset statistics.** # N and # C are the numbers of numerical and categorical columns, respectively.

Out-of-Distribution Public Data. When training DP-2Stage on the Adult dataset, which is considered as the private dataset to be protected, the publicly available Airline dataset is used as the out-distribution pseudo data for training the Stage 1 model in the DP-2Stage-O approach, and similarly for the reverse scenario.

---

[1] `https://huggingface.co/`

[2] `https://opacus.ai/`

[3] `https://github.com/sdv-dev/CTGAN`

[4] `https://github.com/opendp/smartnoise-sdk`

[5] `https://archive.ics.uci.edu/dataset/2/adult`

[6] `https://www.kaggle.com/datasets/teejmahal20/airline-passenger-satisfaction`

Evaluation Metrics. We evaluate the quality of synthetic tabular datasets using two prominent metrics, as detailed in Appendix C, based on the held-out test dataset: Machine Learning Efficacy (utility) and the Normalized Histogram Intersection (fidelity). Machine Learning Efficacy assesses the utility of synthetic data by comparing the performance of machine learning models trained on synthetic data to those trained on real data. The Normalized Histogram Intersection quantifies fidelity by measuring the similarity between the marginal distributions of real and synthetic data, summing the minimum probabilities across corresponding bins, and averaging across all columns. For the utility metric, we train logistic regression[7] and XGB models[8], reporting the average F1-score (**F1**), computed the area under the receiver operating characteristic curve (**AUC**) from prediction scores, and the accuracy score (**ACC**) for each dataset. For the histogram metric (**HIST**), we compute averages using bins of 20 and 50. The individual results for each metrics is shown in Appendix E. Additionally, for language model-based methods, we employ the perplexity metric to evaluate the model's ability to generate accurate next tokens, focusing specifically on value perplexity (**Value Perp**) by masking out the perplexity of the value tokens.

Each model is run five times and four synthetic datasets generated per run. The size of the synthethic data is the same as the training data.

5.2 Non-DP Benchmarks

Table 2 highlights the effectiveness of LLMs using GPT-2 in comparison to other benchmark methods. GPT-2, fine-tuned over 10 epochs, significantly outperforms other models in terms of utility on the Adult dataset and achieves competitive results in terms of the histogram intersection metric. For the Airline dataset, the performance is also competitive compared to models such as CTGAN, TVAE, and VAE. These results clearly demonstrate the strong performance and capability of LLMs in this context. The Real data baseline represents the optimal performance achievable, calculated by evaluating the metrics using the real training data and comparing them against the real test data.

		Adult				Airline			
	Method	**F1**	**AUC**	**ACC**	**HIST**	**F1**	**AUC**	**ACC**	**HIST**
	Real data	$69.9_{\pm 2}$	$91.7_{\pm 1}$	$84.0_{\pm 4}$	$99.1_{\pm 0}$	$90.6_{\pm 8}$	$96.2_{\pm 5}$	$91.8_{\pm 7}$	$99.4_{\pm 0}$
$\varepsilon = \infty$	CTGAN	$59.5_{\pm 6}$	$\underline{88.5}_{\pm 0}$	$80.2_{\pm 3}$	$\underline{91.2}_{\pm 1}$	$\underline{87.2}_{\pm 3}$	$\underline{94.7}_{\pm 2}$	$\underline{88.9}_{\pm 3}$	**94.4**$_{\pm 1}$
	TVAE	$\underline{63.2}_{\pm 2}$	$87.5_{\pm 1}$	$77.7_{\pm 4}$	**91.5**$_{\pm 1}$	$85.8_{\pm 5}$	$93.0_{\pm 6}$	$87.2_{\pm 6}$	$90.3_{\pm 2}$
	VAE	$53.8_{\pm 10}$	$86.6_{\pm 1}$	$\underline{80.5}_{\pm 1}$	$73.3_{\pm 3}$	$79.8_{\pm 1}$	$91.1_{\pm 1}$	$80.0_{\pm 1}$	$76.8_{\pm 0}$
	GPT-2	**68.9**$_{\pm 0}$	**90.7**$_{\pm 0}$	**83.7**$_{\pm 2}$	$90.7_{\pm 1}$	**89.6**$_{\pm 5}$	**95.9**$_{\pm 3}$	**91.4**$_{\pm 4}$	$\underline{90.8}_{\pm 1}$

Table 2: **Non-DP Benchmark.** The Real data baseline represents the optimal achievable performance, determined by evaluating metrics using real training data compared to real test data. Utility metrics are reported as averages, with each model run five times and four synthetic datasets generated per run with standard deviation reported after ±. The best value per column for each ε is shown in **bold** while the second best value is underlined.

5.3 DP Benchmarks

Table 3 presents a comparison of DP-2Stage-O against GAN and VAE-based methods under arguably strict privacy setting of $\varepsilon = \{1, 8\}$. Each model was trained five times, with results averaged across runs and four synthetic datasets generated per run. The evaluation was conducted using tabular data metrics described in section 5.1. Our proposed methods, DP-2Stage-U and DP-2Stage-O, demonstrate competitive performance across all metrics on both datasets. Notably, DP-CTGAN and DP-GAN show greater improvement on the Adult dataset at $\varepsilon = 1$, while DP-2Stage-O outperforms DP-GAN at $\varepsilon = 8$. However, scaling DP-GAN to higher ε values for the Airline dataset proved challenging, requiring up to five days to run before the process was terminated.

---

[7] `https://scikit-learn.org/1.5/modules/generated/sklearn.linear_model.LogisticRegression.html`

[8] `https://xgboost.readthedocs.io/`

		Adult				Airline			
	Method	**F1**	**AUC**	**ACC**	**HIST**	**F1**	**AUC**	**ACC**	**HIST**
$\varepsilon = 1$	DP-GAN	33.5$_{\pm 20}$	67.7$_{\pm 9}$	64.2$_{\pm 10}$	63.7$_{\pm 3}$	40.2$_{\pm 24}$	63.9$_{\pm 13}$	59.8$_{\pm 6}$	44.7$_{\pm 12}$
	DP-CTGAN	**42.2**$_{\pm 20}$	**78.0**$_{\pm 7}$	**75.7**$_{\pm 3}$	75.7$_{\pm 2}$	67.1$_{\pm 8}$	76.7$_{\pm 8}$	68.0$_{\pm 6}$	78.7$_{\pm 2}$
	DP-VAE	0.0$_{\pm 0}$	50.0$_{\pm 0}$	75.6$_{\pm 0}$	61.8$_{\pm 2}$	26.5$_{\pm 28}$	57.9$_{\pm 13}$	57.3$_{\pm 6}$	41.8$_{\pm 1}$
	GPT-2								
	DP-Standard	27.8$_{\pm 15}$	58.5$_{\pm 7}$	65.2$_{\pm 9}$	85.7$_{\pm 2}$	60.5$_{\pm 7}$	65.3$_{\pm 9}$	62.4$_{\pm 7}$	90.3$_{\pm 3}$
	DP-2Stage-U	21.2$_{\pm 12}$	48.9$_{\pm 6}$	61.9$_{\pm 13}$	86.7$_{\pm 1}$	**68.5**$_{\pm 9}$	**77.8**$_{\pm 10}$	**72.1**$_{\pm 7}$	90.7$_{\pm 1}$
	DP-2Stage-O	30.4$_{\pm 17}$	61.6$_{\pm 8}$	66.7$_{\pm 8}$	**88.5**$_{\pm 1}$	55.2$_{\pm 18}$	62.5$_{\pm 19}$	60.0$_{\pm 16}$	**92.5**$_{\pm 1}$
$\varepsilon = 8$	DP-GAN	19.6$_{\pm 20}$	50.0$_{\pm 0}$	50.0$_{\pm 26}$	33.3$_{\pm 9}$	-	-	-	-
	DP-CTGAN	**46.5**$_{\pm 18}$	**79.4**$_{\pm 4}$	73.1$_{\pm 6}$	80.0$_{\pm 2}$	67.7$_{\pm 4}$	76.7$_{\pm 5}$	67.7$_{\pm 4}$	76.8$_{\pm 1}$
	DP-VAE	0.0$_{\pm 0}$	50.0$_{\pm 0}$	**75.6**$_{\pm 0}$	62.1$_{\pm 1}$	51.9$_{\pm 25}$	72.4$_{\pm 10}$	67.2$_{\pm 7}$	40.0$_{\pm 1}$
	GPT-2								
	DP-Standard	31.3$_{\pm 15}$	62.2$_{\pm 7}$	67.7$_{\pm 7}$	84.5$_{\pm 1}$	64.9$_{\pm 6}$	69.8$_{\pm 9}$	65.9$_{\pm 7}$	89.8$_{\pm 3}$
	DP-2Stage-U	22.4$_{\pm 15}$	51.8$_{\pm 8}$	63.7$_{\pm 11}$	86.9$_{\pm 1}$	**71.9**$_{\pm 9}$	**80.7**$_{\pm 10}$	**74.9**$_{\pm 8}$	90.4$_{\pm 1}$
	DP-2Stage-O	33.4$_{\pm 16}$	63.8$_{\pm 9}$	68.2$_{\pm 7}$	**87.9**$_{\pm 1}$	64.2$_{\pm 11}$	71.7$_{\pm 10}$	67.8$_{\pm 8}$	**92.3**$_{\pm 1}$

Table 3: **DP Benchmark.** Utility metrics (F1, AUC, and ACC) are presented as the averages of logistic regression and XGBoost performance. HIST represents the average histogram intersection scores calculated using bins of 20 and 50. Results are averaged across five model runs and four synthetic datasets per run with standard deviation reported after ±. The best value per column for each ε is shown in **bold** while second best value is underlined.

5.4 DP-2Stage Benchmark

In Table 3, we present the results of our proposed DP-2Stage method, which utilizes two pseudo-dataset approaches (DP-2Stage-U and DP-2Stage-O) as described in Section 4.1. The results demonstrate consistent improvements in downstream task utility, with DP-2Stage-O achieving the best accuracy metrics for the Adult dataset and DP-2Stage-U outperforming for the Airline dataset. Furthermore, DP-2Stage-U, leveraging uniform pseudo-data, achieves the fastest inference time—up to 21x faster (see Table 7 in Appendix F)—making it particularly advantageous for real-world deployment.

5.4.1 Impact of Column Shuffling

Next, we evaluate the impact of column shuffling. In Figure 3, we compared the perplexity of DP-2Stage method with DP-Standard. The top section shows the evolution of perplexity with column shuffling enabled, the default in prior work Borisov et al. (2023), while in the bottom we shows perplexity without column shuffling. The results demonstrate that DP-2Stage consistently outperforms DP-Standard in perplexity across both settings, with the non-shuffling configuration yielding the best performance under DP.

In Table 4, we provide a detailed comparison of these two approaches across the evaluated datasets. The results suggest that column shuffling generally benefits the Non-DP setting, which may explains its frequent use in prior research. However, under DP, models without shuffling more often achieve better performance, as indicated by the underlined or bold values, although histogram metrics occasionally favor models with shuffling. Additionally, enabling shuffling significantly increases sampling time (approximately 42-51 hours per model run), particularly for DP-Standard and DP-2Stage-O model (see Appendix 7), as a result, we only present model trained on a single run for DP-Standard and DP-2Stage-O to ensure a fair evaluation when shuffling is enabled.

5.4.2 Impact of Weighted Loss

To evaluate the impact of weighting the loss in the second stage (see Equation 6), we conducted experiments using both the default loss ($\lambda = 0.33$) and the proposed weighting value ($\lambda = 0.65$). The results, presented

			Adult					Airline				
	Method	**Shuffle**	**Value Perp**	**F1**	**AUC**	**ACC**	**HIST**	**Value Perp**	**F1**	**AUC**	**ACC**	**HIST**
GPT-2	Non-DP	✓	2.398±0.00	68.9±0	90.7±0	83.7±2	90.7±1	2.865±0.00	89.6±5	95.9±3	91.4±4	90.8±1
$\varepsilon = \infty$		✗	2.482±0.01	69.0±1	90.7±0	83.3±2	90.2±0	2.901±0.01	76.5±24	91.3±7	84.0±10	93.5±3
	DP-Standard	✓	3.537	24.6±23	**62.4**±8	63.9±14	87.4±0	4.276	44.8±33	73.9±12	66.9±11	93.3±1
		✗	3.224	**31.4**±17	62.1±8	**68.4**±5	85.9±1	3.790	59.4±6	65.8±4	61.8±2	88.3±1
$\varepsilon = 1$	DP-2Stage-U	✓	2.973±0.01	19.7±13	48.0±8	61.9±13	87.6±1	3.932±0.08	48.8±11	57.5±7	55.2±5	**93.4**±1
		✗	**2.887**±0.04	21.2±12	48.9±6	61.9±13	86.7±1	**3.633**±0.01	**68.5**±9	**77.8**±10	**72.1**±7	90.7±1
	DP-2Stage-O	✓	3.270	22.7±22	55.1±10	68.0±9	87.1±1	3.948	59.8±4	68.9±6	66.3±5	92.0±1
		✗	3.049	27.8±18	60.2±7	65.5±9	**88.4**±0	3.737	**73.2**±2	**82.0**±2	**75.7**±2	91.7±1

Table 4: **Comparison of Methods with Shuffle Enabled (✓) or Disabled (✗).** The best result under DP is highlighted in **bold**, while the top-performing result for each shuffle setting is underlined. Non-DP methods generally perform comparably across both settings but tend to show better results with shuffle enabled (✓). Conversely, DP methods often achieve higher performance when shuffle is disabled (✗). Results are averaged across five model runs, each using a different random seed, with four synthetic datasets generated per run. For DP-Standard and DP-2Stage-O, results are based on a single model run, averaging across four synthetic datasets, due to the high inference time required with shuffle enabled (approximately 42–51 hours per run; see Table 7 in Appendix F), ensuring a fair evaluation.

in Table 5, indicate that emphasizing value tokens during the second-stage DP fine-tuning yields the best performance. This approach also improves the performance of DP-Standard, though the improvement is more pronounced for DP-2Stage-O.

Despite the gains observed in DP-Standard when $\lambda = 0.65$, DP-2Stage offers the better performance across both settings. On the Adult dataset, DP-2Stage-U performs slightly better with the default loss in most cases (except for the F1 score), whereas on the Airline dataset, it performs worse with the default loss.

			Adult				Airline			
	Method	λ	**F1**	**AUC**	**ACC**	**HIST**	**F1**	**AUC**	**ACC**	**HIST**
GPT-2	DP-Standard	default	27.8±15	58.5±7	65.2±9	85.7±2	60.5±7	65.3±9	62.4±7	90.3±3
$\varepsilon = 1$		0.65	28.9±17	60.9±8	66.8±8	87.4±2	59.9±7	65.7±9	62.9±7	91.6±4
	DP-2Stage-U	default	20.8±14	49.7±7	63.0±12	87.3±1	65.0±10	74.5±9	69.3±6	91.1±1
		0.65	21.2±12	48.9±6	61.9±13	86.7±1	**68.5**±9	**77.8**±10	**72.1**±7	90.7±1
	DP-2Stage-O	default	30.3±15	60.4±7	65.8±9	87.3±1	48.8±17	55.6±19	54.4±15	92.4±1
		0.65	**30.4**±17	**61.6**±8	**66.7**±8	**88.5**±1	55.2±18	62.5±19	60.0±16	**92.5**±1

Table 5: **Comparison of λ Values for Weighting the Stage 2 Loss.** Results are averaged over five model runs with different random seeds, and four synthetic datasets generated per run.

5.4.3 Impact of Low Perplexity on Utility metrics

In Table 4, we present perplexity values with and without column shuffling alongside utility performance metrics. While uniform pseudo-data (DP-2Stage-U) often achieves lower perplexity, this does not always translate to improved downstream utility or histogram scores. In some cases, the out-distribution pseudo-data (DP-2Stage-O) approach yields better performance. We hypothesize that this discrepancy observed in DP-2Stage-U may result from overfitting when pseudo-data closely resembles the private data. In such cases, the model may lack sufficient incentive to learn meaningful correlations in the private data during the second stage. Conversely, when out-distribution pseudo-data is used, the model is encouraged to learn these correlations while retaining the structural information established during stage 1. This approach may provide better overall performance compared to directly applying DP.

6 Limitations & Future Work

Stage 1 Training Iterations. In our proposed DP-2Stage approach, we trained the stage 1 model for 5 epoch, as done in concurrent work Tran & Xiong (2024). However, further exploration is needed to assess the impact of varying the number of training iterations and identifying an optimal checkpoint during stage one. This could enhance the performance of stage two by providing a more refined initialization point. Future research should also investigate (1) optimal strategies for constructing the stage one dataset, and (2) the influence of the stage one dataset on DP fine-tuning during stage two. These directions highlight important opportunities for advancing the training process.

Serialization Methods. The serialization method employed in this work, based on the approach proposed by Borisov et al. (2023), has not been compared with alternative methods. Exploring different serialization strategies could reveal approaches that are better suited for DP, potentially leading to enhanced performance. Future research could investigate the impact of various serialization methods to identify those that most effectively address the needs of DP.

DP Hyperparameters Tuning. This work did not involve an extensive exploration of hyperparameters, such as batch size, learning rate schedulers, learning rate, or clipping strategies, which are critical for DP performance. Conducting a systematic analysis of these hyperparameters could yield valuable insights into optimizing LLMs for use as DP tabular data generators. Future research in this direction could help unlock the full potential of LLMs in this domain.

Scaling to Larger Models. Exploring the use of larger models presents a promising avenue for enhancing performance. However, this work did not investigate this direction due to the significant computational costs associated with training and inference for models with a high number of parameters. Balancing the trade-off between performance gains and computational feasibility remains a critical challenge. Future research could focus on developing strategies to mitigate these costs, enabling a more practical evaluation of larger models.

Incorporating Additional Datasets. The results presented in our work are based on two datasets, Adult and Airline, which differ in domain, column size, and sample size. Expanding the analysis to include a wider variety of datasets would allow for a more thorough evaluation and enhance the validity of the claims made in this paper. However, due to the computational cost associated with training and inference, we leave this for future work.

Metric Diversity. Our evaluation primarily used perplexity to assess the language model's performance, machine learning efficacy to measure the downstream task utility, and Histogram Intersection to evaluate the synthetic data fidelity. While these metrics are informative and widely adopted in synthetic tabular data research (Xu et al., 2019; Afonja et al., 2023), they do not provide a complete picture of synthetic data quality. Incorporating domain-specific metrics could offer deeper insights into the data's applicability and reveal limitations not captured by general metrics (Chen et al., 2024).

7 Conclusion

In this work, we investigated the use of pre-trained LLMs for tabular data generation under DP protection. Our analysis indicates that naïvely fine-tuning the model results in sub-optimal performance due to inefficient allocation of privacy budgets. In light of this, we propose DP-2Stage, a two-stage optimization approach that first adapt the model to the format of the task in the first stage and proceeds with fine-tuning the model for the DP task. This two-stage strategy secures robust privacy protection while ensuring that privacy budgets are spent on the actual sensitive data, leading to improved data utility and efficient data generation. Despite the competitive performance and increased efficiency, further investigation is needed to optimize privacy budget allocation and improve scalability, as discussed in section 6.

Broader Impact Statement

This research examines the privacy risks of using pre-trained LLMs, such as GPT-2, for generating tabular data and introduces a differentially private fine-tuning process to address these concerns. While this approach reduces re-identification and data leakage by incorporating DP into the training process, it also incurs significant environmental costs due to the high computational demands. For example, training a non-differentially private CTGAN for 300 epochs takes around 10 minutes, whereas fine-tuning GPT-2 for 10 epochs requires approximately 2 hours on the same hardware and dataset, with DP training extending the time further due to per-example gradient computations. This disparity underscores the greater computational expense of LLMs for both training and inference compared to traditional GAN-based tabular data generators. We hope to inspire dialogue on how to leverage the capabilities of LLMs for tabular generation tas in a more sustainable and environmentally conscious manner.

Reproducibility Statement

To ensure reproducibility, we outline several key efforts throughout the paper. The methodology for our proposed DP-2Stage framework is detailed in section 4, and section 5 describes the public datasets and open-source model used in our experiments. Additionally, we have released the source code to enable further experimentation and validation of our findings `https://anonymous.4open.science/r/DP-2Stage-B8B1/README.md`.

References

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 ACM Conference on Computer and Communications Security (CCS)*, 2016.

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Tejumade Afonja, Dingfan Chen, and Mario Fritz. Margctgan: A "marginally" better ctgan for the low sample regime. In *German Conference on Pattern Recognition (GCPR)*, 2023.

Laura Aviñó, Matteo Ruffini, and Ricard Gavaldà. Generating synthetic but plausible healthcare record datasets. *KDD workshop on Machine Learning for Medicine and Healthcare*, 2018.

Borja Balle, Gilles Barthe, Marco Gaboardi, Justin Hsu, and Tetsuya Sato. Hypothesis testing interpretations and renyi differential privacy. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2020.

Vadim Borisov, Kathrin Sessler, Tobias Leemann, Martin Pawelczyk, and Gjergji Kasneci. Language models are realistic tabular data generators. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2023.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Dingfan Chen, Ning Yu, Yang Zhang, and Mario Fritz. Gan-leaks: A taxonomy of membership inference attacks against generative models. In *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*, 2020.

Dingfan Chen, Marie Oestreich, Tejumade Afonja, Raouf Kerkouche, Matthias Becker, and Mario Fritz. Towards biologically plausible and private gene expression data generation. *Proceedings on Privacy Enhancing Technologies*, 2024.

Edward Choi, Siddharth Biswal, Bradley Malin, Jon Duke, Walter F Stewart, and Jimeng Sun. Generating multi-label discrete patient records using generative adversarial networks. In *Machine learning for healthcare conference*, 2017.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. Scaling instruction-finetuned language models. 2022.

Cynthia Dwork. Differential privacy. In *International colloquium on automata, languages, and programming*, 2006.

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

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in Neural Information Processing Systems (NeurIPS)*, 2014.

Benjamin Hilprecht, Martin Härterich, and Daniel Bernau. Monte carlo and reconstruction membership inference attacks against generative models. *Proc. Priv. Enhancing Technol.*, 2019.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 1997.

Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.

Frederick Jelinek. *Statistical methods for speech recognition.* MIT press, 1998.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2014.

Akim Kotelnikov, Dmitry Baranchuk, Ivan Rubachev, and Artem Babenko. Tabddpm: Modelling tabular data with diffusion models. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2023.

Ryan McKenna, Brett Mullins, Daniel Sheldon, and Gerome Miklau. Aim: an adaptive and iterative mechanism for differentially private synthetic data. *Proceedings of the VLDB Endowment*, 2022.

Ilya Mironov. Rényi differential privacy. In *2017 IEEE 30th computer security foundations symposium (CSF)*. IEEE, 2017.

Ilya Mironov, Kunal Talwar, and Li Zhang. Rényi differential privacy of the sampled gaussian mechanism. *arXiv preprint arXiv:1908.10530*, 2019.

Noseong Park, Mahmoud Mohammadi, Kshitij Gorde, Sushil Jajodia, Hongkyu Park, and Youngmin Kim. Data synthesis based on generative adversarial networks. *Proc. VLDB Endow.*, 2018.

Natalia Ponomareva, Hussein Hazimeh, Alex Kurakin, Zheng Xu, Carson Denison, H Brendan McMahan, Sergei Vassilvitskii, Steve Chien, and Abhradeep Guha Thakurta. How to dp-fy ml: A practical guide to machine learning with differential privacy. *Journal of Artificial Intelligence Research*, 77:1113–1201, 2023.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 2019.

Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*, 2015.

Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE symposium on security and privacy (SP)*, pp. 3–18. IEEE, 2017.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2015.

Rubèn Tito, Khanh Nguyen, Marlon Tobaben, Raouf Kerkouche, Mohamed Ali Souibgui, Kangsoo Jung, Joonas Jälkö, Vincent Poulain D'Andecy, Aurelie Joseph, Lei Kang, et al. Privacy-aware document visual question answering. In *International Conference on Document Analysis and Recognition*, 2024.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

Toan V Tran and Li Xiong. Differentially private tabular data synthesis using large language models. *arXiv preprint arXiv:2406.01457*, 2024.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Hui-Po Wang and Mario Fritz. Language models as zero-shot lossless gradient compressors: Towards general neural parameter prior models. *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. Finetuned language models are zero-shot learners. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.

Liyang Xie, Kaixiang Lin, Shu Wang, Fei Wang, and Jiayu Zhou. Differentially private generative adversarial network. *arXiv preprint arXiv:1802.06739*, 2018.

Lei Xu, Maria Skoularidou, Alfredo Cuesta-Infante, and Kalyan Veeramachaneni. Modeling tabular data using conditional gan. *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Da Yu, Saurabh Naik, Arturs Backurs, Sivakanth Gopi, Huseyin A Inan, Gautam Kamath, Janardhan Kulkarni, Yin Tat Lee, Andre Manoel, Lukas Wutschitz, et al. Differentially private fine-tuning of language models. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.

Jun Zhang, Graham Cormode, Cecilia M Procopiuc, Divesh Srivastava, and Xiaokui Xiao. Privbayes: Private data release via bayesian networks. *ACM Transactions on Database Systems (TODS)*, 2017.

Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. Large language models are human-level prompt engineers. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2022.