IS API ACCESS TO LLMS USEFUL FOR GENERATING PRIVATE SYNTHETIC TABULAR DATA?

Anonymous authors
Paper under double-blind review

ABSTRACT

Differentially private (DP) synthetic data is a versatile tool for enabling the analysis of private data. Recent advancements in large language models (LLMs) have inspired a number of algorithm techniques for improving DP synthetic data generation. One family of approaches uses DP finetuning on the foundation model weights; however, the model weights for state-of-the-art models may not be public. In this work we propose two DP synthetic tabular data algorithms that only require API access to the foundation model. We adapt the Private Evolution algorithm (Lin et al., 2023; Xie et al., 2024)—which was designed for image and text data—to the tabular data domain. In our extension of Private Evolution, we define a query workload-based distance measure, which may be of independent interest. We propose a family of algorithms that use one-shot API access to LLMs, rather than adaptive queries to the LLM. Our findings reveal that API-access to powerful LLMs does not always improve the quality of DP synthetic data compared to established baselines that operate without such access. We provide insights into the underlying reasons and propose improvements to LLMs that could make them more effective for this application.

1 INTRODUCTION

Synthetic data has long been a “holy grail” for performing computations on sensitive data, with the allure of protecting privacy while supporting typical data queries and regular data workflows out-of-the-box. Unfortunately, without a rigorous treatment of privacy, the synthetic dataset may inadvertently reveal information about the sensitive data from which it is derived.

Differential privacy (DP) (Dwork et al., 2006; 2016) has emerged as the gold standard for quantifying privacy leakage by algorithms that process sensitive data records from users. At a high level, a (randomized) algorithm satisfies differential privacy if the algorithm’s output distribution is not affected very much by a single person’s data, regardless of what the other data records are. This ensures that the mechanism’s output reveals little about any individual person’s data as a result of their participation in the data analysis, even after arbitrary post-processing of the mechanism output.

Many algorithms have been developed for DP synthetic data, particularly for tabular data (McKenna et al., 2022; Tao et al., 2021; Liu et al., 2021b; Aydore et al., 2021; Liu et al., 2021a; Cai et al., 2021; Zhang et al., 2021). With the advancement of large language models (LLMs), a number of recent works propose improved DP synthetic data algorithms that use LLMs trained on public data[1]. Among these are two broad categories of methods: those which privately finetune a foundation model, and those which only use API access to the foundation model. Sablayrolles et al. (2023), Tran & Xiong (2024), and Afonja et al. (2024) use private finetuning on generative language models to generate private tabular synthetic data, and Kurakin et al. (2023) similarly do private LoRA finetuning on an LLM to generate synthetic text data. Similarly, Ghalebikesabi et al. (2023) employ DP finetuning of diffusion models for generating DP synthetic images.

Despite their power, these DP finetuning methods have significant hurdles. First, finetuning algorithms require white-box access to the model, as the weights need to be directly adjusted. This is a

[1]The extent to which data used to train LLMs is considered *public* and compatible with privacy goals is hotly contested (Tramèr et al., 2022). We sidestep this question and assume public models are fair to treat as non-private, but we acknowledge it remains an important question.

problem because many state-of-the-art models are proprietary, with weights that remain confidential. Only a limited set of researchers are able to even experiment with DP finetuning on such models. Secondly, the resources needed for DP finetuning scales with model dimensionality; time and energy costs quickly become prohibitive. These hurdles motivate alternative ways of using foundation models. In particular, even many proprietary models have a publicly accessible API.

A series of works in the synthetic image (Lin et al., 2023) and text (Xie et al., 2024) domains use only API access to foundation models. The algorithm, Private Evolution, combines adaptive queries to the foundation model with a genetic algorithm to privately generate synthetic image and text data. These methods were further extended to the federated setting by Hou et al. (2024). Yet a different approach (Amin et al., 2024) uses private prediction combined with other privacy budget saving tricks on the foundation model to generate DP synthetic text.

In light of these recent successes for image and text data, we ask: *Can API access to an LLM improve algorithms for generating DP synthetic **tabular** data?*

A priori it isn't obvious why an LLM would be useful at all for generating synthetic tabular data that it wasn't trained on; however, in initial experiments we found that with descriptive column names, the LLM we used has a reasonable prior over realistic-looking data records. This prior is a powerful source of information we harness in our algorithms.

We design and evaluate two types of DP synthetic tabular generation algorithms that leverage LLM API access. In Section 2, we adapt Private Evolution (Lin et al., 2023; Xie et al., 2024) to the tabular domain. A key part of our solution uses a workload-aware family of distance functions, which may be of independent interest, to align the genetic algorithm with the final workload error. In Section 3 we introduce a new class of private synthetic data algorithms that use one-shot API access to the foundation model. Unlike prior methods, which require adaptive queries to the foundation model or finetuning the model's weights, our method consumes only one (offline) round of queries to the foundation model. Along the way, we evaluate our two approaches against a number of accuracy baselines to determine whether they advance the state-of-the-art for DP synthetic tabular data.

We evaluated our algorithms with Gemini 1.0 Pro (Gemini Team Google, 2023), which allowed us to constrain the outputs to structured tabular records. In our evaluations, we find that the proposed methods fail to consistently beat our baselines. Despite this, we think our attempts are instructive to the research community and could inform the development of state-of-the-art methods, especially as foundation models improve. In light of our findings, we share our key take-aways:

The role of data domain. The state-of-the-art for DP synthetic data generation is highly domain specific. In particular, DP tabular synthetic data has been very well-studied compared to image and text, so the state of the art for tabular data is much harder to improve on. Additionally, prior work on Private Evolution relies on public image and text embeddings to measure the fidelity of the synthetic data, but similar embeddings do not exist for tabular data. Our workload-aware distance function in Section 2 is one substitute, but surely other solutions exist as well.

The importance of appropriate baselines. In the tabular data domain, there is no single algorithm that dominates on all datasets, query workloads, and privacy budgets. Any new algorithm in this area requires extensive comparison to the handful of algorithms that dominate the state-of-the-art, as well as naive baselines. In Section 3, we show that combining Gemini-generated data with JAM (Fuentes et al., 2024) outperforms all other methods; however, in testing other baselines we find that this holds regardless of the public data we give JAM. Without this naive baseline, we would have reached a false conclusion that the Gemini-generated data was improving the state-of-the-art.

2 Adapting Private Evolution to Tabular Data

We adapt Private Evolution (PE) (Lin et al., 2023; Xie et al., 2024) to the tabular data domain. Private Evolution works in rounds, by maintaining a set of *candidates* S_t generated by the foundation model and using a distance function together with a differentially private histogram to have each private record individually vote for candidates. The best performing synthetic candidates become part of an *elite set* for that round S_t'; at the end of each round, the foundation model is prompted to generate more examples similar to the elite set, which then become the new candidates S_{t+1}.

The first set of candidates are populated by a $\mathsf{Random\_API}$, which prompts the model to generate some prespecified number of initial candidates adhering to the column names and datatypes of the private dataset. Each subsequent set of candidates are generated via the $\mathsf{Variation\_API}$ which takes the current elite set of candidates and prompts the model to generate some number of additional candidates that are similar.

Algorithm 1 Private Evolution (Lin et al., 2023; Xie et al., 2024)

Input: Private samples S_{priv}, Number of iterations T, Number of generated samples N_{synth}, Distance function $dist_\varepsilon(\cdot,\cdot)$, Noise multiplier σ
Output: Synthetic data S_{synth}

$S_1 \leftarrow \mathsf{Random\_API}(2 \cdot N_{\mathsf{synth}})$
for $t = 1$ to T **do**
  $H = []$ ▷ Initialize histogram over S_t
  for $x_{priv} \in S_{\mathsf{priv}}$ **do**
    $i = \arg\min_{j\in[n]} dist_\varepsilon(x_{priv}, S_t)$ ▷ Compute closest synthetic candidate
    $H[i] = H[i] + 1$
  $H \leftarrow H + \mathcal{N}(0, \sigma I_{2\cdot N_{\mathsf{synth}}})$ ▷ Add noise to ensure DP
  $H \leftarrow \max(0, H)$ ▷ Post-process element-wise
  $\mathcal{P}_t \leftarrow H/sum(H)$ ▷ Compute empirical distribution on S_t
  $S'_t \leftarrow$ draw N_{synth} samples with replacement from $\mathcal{P}_t$
  $S_{t+1} \leftarrow \mathsf{Variation\_API}(S'_t)$
return S_T

2.1 Workload-Aware Distance Function

Prior methods that applied Private Evolution to image (Lin et al., 2023) and text (Xie et al., 2024) data used public text and image embeddings, respectively, to measure the distance between candidate synthetic examples and the private examples. Choosing a sensible distance function for tabular records is less straightforward: public tabular embeddings (if they exist) likely wouldn't capture the features of unseen data, simple ℓ_p distance fails to account for differences in scale among columns.

Instead, we derive a workload-aware distance function. A private synthetic dataset is typically optimized for and evaluated on a particular *workload* of (linear) queries $W = \{q_1, \ldots, q_k\}$. The workload error is typically some ℓ_p variation on:

$$\mathsf{WError}(S_{\mathsf{priv}}, S_{\mathsf{synth}}) = \sum_{i\in[k]} |q_i(S_{\mathsf{priv}}) - q_i(S_{\mathsf{synth}})|.$$

Note that workload error is a function of pairs of datasests; however, the distance function we require is a function of pairs of individual records. We unpack the workload error further: assuming the queries are *linear*, then they correspond to a sum over a predicate on data records $q_i(\mathbf{x}) = \sum_{j\in[n]} \psi_i(x_j)$. Thus, for the given predicates $\psi = (\psi_1, \ldots, \psi_k)$ corresponding to the queries in W, we will define the workload-aware distance function between a private record and synthetic candidate:

$$\mathsf{Wdist}_\psi(x, c) = \sum_{i\in k} |\psi_i(x) - \psi_i(c)|.$$

A dataset of synthetic candidates with low workload-aware distance will have low workload error compared to the private data.

2.2 Experimental results

We evaluate our adapted private evolution algorithm on a modified version of NYC Taxi and Limousine Commission data using Gemini 1.0 Pro. We use data from January 2024, to to avoid data contamination between the public and private evaluation data (Google Cloud, 2023).

For our workload, we use a scaled ℓ_1-distance on numerical variables for each combination of categorical variables. We rescale the numerical variables to account for different value ranges—for

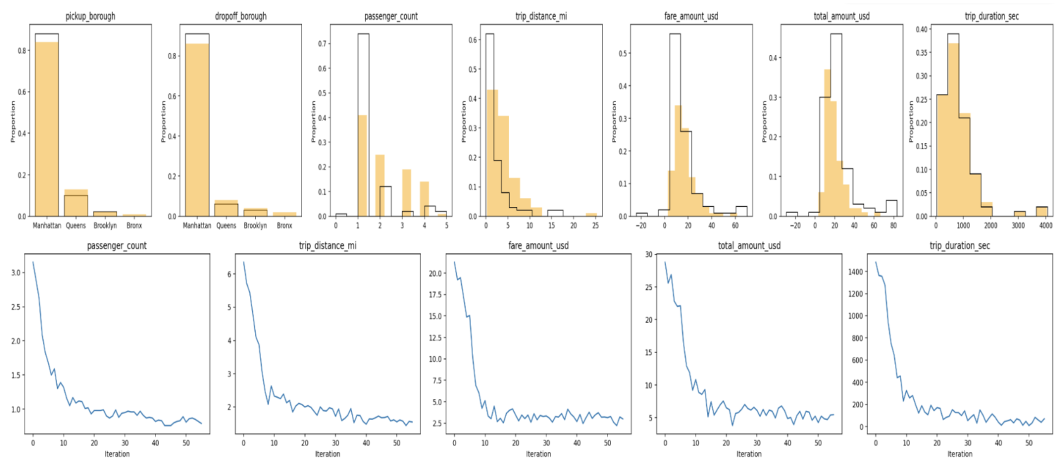

Figure 1: **Top** 1-way marginals on private (outlined) and synthetic (yellow) data. **Bottom** workload error of synthetic data over time for Private Evolution with $\varepsilon = \infty$.

example, trip distance (in miles) versus trip duration (in seconds). As an initial experiment, we ran the algorithm without any privacy constraints, and we found that the workload error converges and the 1-way marginals of the synthetic data converges to the 1-way marginals on the private data as well (see Figure 1). It's worth noting that, depending on the expressivity of the foundation model, it's not a given that the PE algorithm would converge even without privacy.

We then ran the same experiments but with a DP histogram instead of a nonprivate one. We experimented with various hyperparameters: how many initial random examples to use, how many iterations to use, how to split the budget across iterations, etc. We found that using increasing budget across runs worked better than an even or decreasing budget; additionally, having more candidates relative to N_{synth} worked best, and finally, using fewer iterations worked best—in fact, using *only a single iteration* was the optimal setting we could find.

With differential privacy, the private evolution algorithm failed to beat two simple baselines: *independent* which privately computes all 1-way marginals and samples data from the product over the private marginals, and *DP workload* which directly computes the workload queries with DP, without generating any synthetic data. These two baselines are not the only two which we'd hope to beat, rather, they're the bare minimum. Moreover, querying any large foundation model hundreds of times is relatively slow.

What we learned The observation that the workload-aware private evolution algorithm performs best with *one shot* data generation implies that: *whatever marginal gains we get from iterating multiple times, they are outweighed by the privacy cost of composing over iterations.* Additionally, while PE was developed for image and text domains where finetuning a foundation model is the main alternative for DP synthetic data, there is a vast literature on algorithms for private synthetic *tabular* data that do not require access to generative models. These lessons paved the way for our second attempt, which proved to be more successful, though still did not beat the current state-of-the-art.

3 Using Gemini-Generated Records as Public Data

There is a substantial body of work on DP synthetic tabular data. Some state-of-the-art algorithms within this space make use of *public data* to improve the accuracy or efficiency of the algorithm on private data; however, for many applications such public data may not be available in the format required[2], as is discussed in-depth in Liu et al. (2021b)[Section 6.1]. Our second approach uses Gemini generated data in lieu of this public data.

[2]This is especially true for algorithms that assume the public dataset has the same (or substantially overlapping) columns, or even is distributed similarly to the private data.

3.1 APPROACH OVERVIEW

Using Gemini's structured output functionality, we prompt Gemini to generate data records with a response schema matching the column names and datatypes of our private dataset. Importantly, none of the private records influence the prompts to Gemini—only the column names and datatypes do. This data generation occurs "offline" and in one shot with no loss of privacy budget. We call this dataset $\mathsf{Gem\_Synth}$. Later, we plug this synthetic public dataset into various DP synthetic tabular algorithms that use public data as well as the private data.

One major benefit of the one-shot nature of this method (rather than querying Gemini interactively in a loop) is we can generate *many* synthetic public data records and reuse the same generated records when trying different approaches. This is not possible when the records are generated adaptively as in Private Evolution. Thus, this method takes advantage of our observations about PE. We begin with a high-level overview of how public data is incorporated into two DP synthetic data algorithms. For both, we consider what happens when we use Gemini-generated tabular data as the public data source for these algorithms.

$\mathsf{PMW}^{\mathsf{pub}}$ Liu et al. (2021a) The $\mathsf{PMW}^{\mathsf{pub}}$ algorithm is an improvement of MWEM (Hardt et al., 2012), which we will not discuss in detail. The basic idea is to use public data to initialize the generating distribution over synthetic records and iteratively refine this distribution to reduce the workload error. The public records reduce the number of iterations required, by providing a "warm start" for the synthetic data distribution, along with reducing the data domain over which the distribution is estimated.

A key sub-routine of $\mathsf{PMW}^{\mathsf{pub}}$ is to estimate a distribution that approximately matches some noisy statistics. Specifically, let Q denote a collection of linear queries and let $\tilde{y} = Q(D) + \xi = \sum_{x \in D} Q(x) + \xi$ be the noisy answers to those queries on the sensitive data. $\mathsf{PMW}^{\mathsf{pub}}$ finds a distribution supported on the "public" data $\mathsf{Gem\_Synth}$, and finds the weights to assign to each public record to minimize the ℓ_2 squared error to the noisy observations.

$$w^* = \operatorname*{arg\,min}_{w \in \mathbb{R}_+} \left\| \sum_{x \in \mathsf{Gem\_Synth}} w_x Q(x) - \tilde{y} \right\|_2^2$$

When the public records are sufficiently representative, this method can work quite well. However, with small or unrepresentative public datasets, this method may not find a good distribution even in the absence of noise.

Gemini inference We use the Gemini-generated records as the public records for the subroutine of $\mathsf{PMW}^{\mathsf{pub}}$, calling this "Gemini inference", setting Q to be the query workload.

MST modified to take public data The standard MST algorithm (McKenna et al., 2021) has three phases: selecting marginal queries, measuring the marginals with DP, and lastly using Private-PGM to post-process the noisy marginals and generate a synthetic dataset. We modify the final step (generation), replacing Private-PGM with the subroutine from $\mathsf{PMW}^{\mathsf{pub}}$ that utilizes $\mathsf{Gem\_Synth}$. This method differs from the Gemini inference approach primarily in how the queries Q are selected.

JAM The JAM-PGM mechanism (Fuentes et al., 2024) was developed for marginal queries, and utilizes public data in a different manner. It privately decides whether to measure each marginal query on the public data or the private data in order to minimize the overall workload error. This mechanism has the benefit that it can utilize public data that is accurate on some, but not necessarily all, marginals. We run this mechanism as-is, using $\mathsf{Gem\_Synth}$ as the public data.

3.2 BASELINES FOR COMPARISON

Because there are a wealth of methods for generating private synthetic data with and without public data, we have a fair number of baselines that we need to compare any new methods to. A number of works that privately finetune foundation models for tabular data omit comparisons to state-of-the-art methods for generating DP tabular data, so it is unclear if they outperform existing approaches.

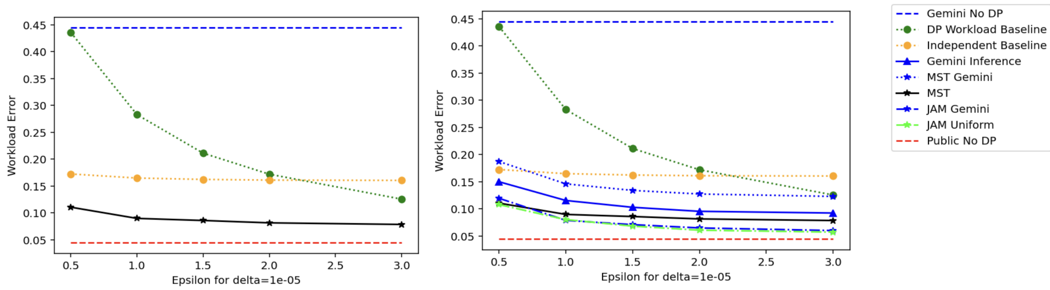

Figure 2: **(Left)** Workload error for baseline methods for generating tabular synthetic data without use of Gemini. **(Right)** Workload error for baseline methods and our one-shot methods that use API access to Gemini.

We study two baselines that require no privacy. First, we consider an **in-distribution public dataset:** publically data drawn from the same distribution as the private data. This is essentially the lowest error we could hope for, up to sampling error; however, in-distribution public data is usually not available. Second, we consider using the **Gemini data with no DP** to answer workload queries.

Next, we consider a number of baselines that do not require public data. **DP workload**: we compute the queries directly using DP. **Independent baseline**: privately compute the 1-way marginals and sample records from the corresponding product distribution over marginals. **MST algorithm**: a tabular synthetic data algorithm that does not use public data. $\mathsf{PMW}^{\mathsf{pub}}$ **with uniform data** and **JAM with uniform data**: using data that is drawn uniformly from the domain as public data for these algorithms.

3.3 Experimental Setup

Our private dataset is UCI Adult (Becker & Kohavi, 1996). Using this structured output constraint, we sample Gemini with top-k=1 and temperature=1 to generate 131,000 records in Gem_Synth. We use 2-way marginals as our query workload to evaluate the fidelity of the DP synthetic data.

3.4 Results

Figure 2 (Left) shows the workload error versus epsilon for the baseline methods discussed. Among these methods, MST achieves the lowest workload error (except the in-distribution public dataset which is our unachievable "best-case baseline"). Figure 2 (Right) shows all of the results for the baseline methods in addition to the methods that use Gem_Synth. Notice that JAM with Gem_Synth performs best overall; however, JAM performs equally well with uniform data. This is because JAM is simply using the private data to compute answers to the queries rather than utilizing the public data. Thus, JAM with Gem_Synth is not better than the state-of-the-art methods on this dataset and query workload. In general, Gem_Synth may capture 1-way marginals on the data better than uniform, however it is generally inaccurate on k-way marginals.

4 Conclusion and Future Work

We evaluated two methods for incorporating API access to Gemini for generating DP synthetic tabular data. While our methods did not beat state-of-the-art methods, this work motivates a number of future directions. First, as foundation models continue to improve, combining our methods with better models (e.g. models trained on more tabular data) could potentially improve the final accuracy, especially if the models are trained specifically for the tabular setting. Additionally, because Gemini uses word embeddings, perhaps doing some finetuning on publicly available tabular data could improve the quality of the Gemini-generated tabular records fed into our one-shot method. Lastly, perhaps there are ways to achieve better accuracy by combining Private Evolution and our one-shot approach. Using foundation models for DP synthetic data generation is still a very new area of research, with many avenues for improvements and breakthroughs.

References

Tejumade Afonja, Hui-Po Wang, Raouf Kerkouche, and Mario Fritz. Dp-2stage: Adapting language models as differentially private tabular data generators, 2024. URL `https://arxiv.org/abs/2412.02467`.

Kareem Amin, Alex Bie, Weiwei Kong, Alexey Kurakin, Natalia Ponomareva, Umar Syed, Andreas Terzis, and Sergei Vassilvitskii. Private prediction for large-scale synthetic text generation. *arXiv preprint arXiv:2407.12108*, 2024.

Sergul Aydore, William Brown, Michael Kearns, Krishnaram Kenthapadi, Luca Melis, Aaron Roth, and Ankit A Siva. Differentially private query release through adaptive projection. In *International Conference on Machine Learning*, pp. 457–467. PMLR, 2021.

Barry Becker and Ronny Kohavi. Adult. UCI Machine Learning Repository, 1996. DOI: https://doi.org/10.24432/C5XW20.

Kuntai Cai, Xiaoyu Lei, Jianxin Wei, and Xiaokui Xiao. Data synthesis via differentially private markov random fields. *Proceedings of the VLDB Endowment*, 14(11):2190–2202, 2021.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*, pp. 265–284. Springer, 2006.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. *Journal of Privacy and Confidentiality*, 7(3):17–51, 2016.

Miguel Fuentes, Brett C Mullins, Ryan McKenna, Gerome Miklau, and Daniel Sheldon. Joint selection: Adaptively incorporating public information for private synthetic data. In *International Conference on Artificial Intelligence and Statistics*, pp. 2404–2412. PMLR, 2024.

Gemini Team Google. Gemini: A family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

Sahra Ghalebikesabi, Leonard Berrada, Sven Gowal, Ira Ktena, Robert Stanforth, Jamie Hayes, Soham De, Samuel L Smith, Olivia Wiles, and Borja Balle. Differentially private diffusion models generate useful synthetic images. *arXiv preprint arXiv:2302.13861*, 2023.

Google Cloud. Generative AI on Vertex AI Documentation: Google Models, 2023. URL `https://cloud.google.com/vertex-ai/generative-ai/docs/learn/models#gemini-1.0-pro`.

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

Charlie Hou, Akshat Shrivastava, Hongyuan Zhan, Rylan Conway, Trang Le, Adithya Sagar, Giulia Fanti, and Daniel Lazar. Pre-text: Training language models on private federated data in the age of llms. *arXiv preprint arXiv:2406.02958*, 2024.

Alexey Kurakin, Natalia Ponomareva, Umar Syed, Liam MacDermed, and Andreas Terzis. Harnessing large-language models to generate private synthetic text. *arXiv preprint arXiv:2306.01684*, 2023.

Zinan Lin, Sivakanth Gopi, Janardhan Kulkarni, Harsha Nori, and Sergey Yekhanin. Differentially private synthetic data via foundation model apis 1: Images. *arXiv preprint arXiv:2305.15560*, 2023.

Terrance Liu, Giuseppe Vietri, Thomas Steinke, Jonathan Ullman, and Steven Wu. Leveraging public data for practical private query release. In *International Conference on Machine Learning*, pp. 6968–6977. PMLR, 2021a.

Terrance Liu, Giuseppe Vietri, and Steven Z Wu. Iterative methods for private synthetic data: Unifying framework and new methods. *Advances in Neural Information Processing Systems*, 34: 690–702, 2021b.

Ryan McKenna, Gerome Miklau, and Daniel Sheldon. Winning the nist contest: A scalable and general approach to differentially private synthetic data. *arXiv preprint arXiv:2108.04978*, 2021.

Ryan McKenna, Brett Mullins, Daniel Sheldon, and Gerome Miklau. Aim: An adaptive and iterative mechanism for differentially private synthetic data. *arXiv preprint arXiv:2201.12677*, 2022.

Alexandre Sablayrolles, Yue Wang, and Brian Karrer. Privately generating tabular data using language models. *arXiv preprint arXiv:2306.04803*, 2023.

Yuchao Tao, Ryan McKenna, Michael Hay, Ashwin Machanavajjhala, and Gerome Miklau. Benchmarking differentially private synthetic data generation algorithms. *arXiv preprint arXiv:2112.09238*, 2021.

Florian Tramèr, Gautam Kamath, and Nicholas Carlini. Position: Considerations for differentially private learning with large-scale public pretraining. In *Forty-first International Conference on Machine Learning*, 2022.

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

Giuseppe Vietri, Cedric Archambeau, Sergul Aydore, William Brown, Michael Kearns, Aaron Roth, Ankit Siva, Shuai Tang, and Steven Z Wu. Private synthetic data for multitask learning and marginal queries. *Advances in Neural Information Processing Systems*, 35:18282–18295, 2022.

Chulin Xie, Zinan Lin, Arturs Backurs, Sivakanth Gopi, Da Yu, Huseyin A Inan, Harsha Nori, Haotian Jiang, Huishuai Zhang, Yin Tat Lee, et al. Differentially private synthetic data via foundation model apis 2: Text. *arXiv preprint arXiv:2403.01749*, 2024.

Mengmeng Yang, Chi-Hung Chi, Kwok-Yan Lam, Jie Feng, Taolin Guo, and Wei Ni. Tabular data synthesis with differential privacy: A survey. *arXiv preprint arXiv:2411.03351*, 2024.

Zhikun Zhang, Tianhao Wang, Ninghui Li, Jean Honorio, Michael Backes, Shibo He, Jiming Chen, and Yang Zhang. {PrivSyn}: Differentially private data synthesis. In *30th USENIX Security Symposium (USENIX Security 21)*, pp. 929–946, 2021.

A PRELIMINARIES

We begin by presenting the definition of differential privacy, which is a constraint on an algorithm $\mathcal{A}$ that processes a dataset $\mathbf{x} = (\mathbf{x}_1, \dots, \mathbf{x}_n)$ of user records, one per user. Two datasets are called *neighbors* if they differ on one person's record. At a high level, differential privacy requires that for any pair of neighboring datasets, the algorithm's output distributions are similar when run on each dataset.

Definition 1 (Differential Privacy (Dwork et al., 2006; 2016)) *A randomized algorithm* $\mathcal{A} : \mathcal{U}^n \rightarrow \mathcal{Y}$ *is* ε-differentially private *if for every pair of neighboring datasets* $\mathbf{x}, \mathbf{x}' \in \mathcal{U}^n$*, and for all outputs* $y \in \mathcal{Y}$*,*

$$\Pr[\mathcal{A}(\mathbf{x}) = y] \leq e^{\varepsilon} \cdot \Pr[\mathcal{A}(\mathbf{x}') = y] + \delta,$$

where the probability is taken over the internal coins of $\mathcal{A}$*.*

The differential privacy guarantee is parameterized by $\varepsilon > 0$, where algorithms with lower values have less privacy leakage and higher values of epsilon denote more privacy leakage from the algorithm's output. DP gives a worst-case guarantee (over the algorithm's inputs and outputs) on how much information an algorithm leaks about its input.

A.1 PRIOR WORK

GAN-based methods for DP synthetic data Many prior works have proposed synthetic data mechanisms based on generative adversarial networks. See Yang et al. (2024) for a nice survey of these and other approaches. These mechanisms generally work by fitting the parameters of the model via DP-SGD, and then using the model to generate synthetic data after training. These techniques are typically best suited for unstructured data like images or text.

Marginal-based methods for DP synthetic data Many mechanisms for DP synthetic data generation work by adding noise to low-dimensional marginals of the data distribution McKenna et al. (2021; 2022); Cai et al. (2021); Aydore et al. (2021); Fuentes et al. (2024); Vietri et al. (2022); Liu et al. (2021b;a); Zhang et al. (2021). Some mechanisms in this space are also designed to leverage public data when it's available Fuentes et al. (2024); Liu et al. (2021b;a). Benchmarks have confirmed these approaches work very well in tabular data settings Tao et al. (2021).