One-Run Privacy Auditing for Structured Generative and Foundation Models

Rishav Chourasia [1 2] **Zilong Zhao** [1 2] **Uzair Javaid** [2]

Abstract

Generative models are gaining traction in synthetic data generation but see limited industry adoption because of lack of standardized data utility, fidelity, and especially privacy metrics. In this paper, we focus on privacy and propose a practical ϵ-differential privacy auditing technique focused on structured generative and foundation models that measures memorization via nearest-neighbor distances between real training data and generated synthetic samples. By independently selecting a small subset of training data for auditing, our method operates in a single training run and treats the generative pipeline as a black box. Our approach models synthetic samples as reconstruction attacks and yields significantly stronger lower bounds on privacy loss than traditional membership inference attacks. We test our technique on five tabular generative models and one foundation model, and show our method provides a robust baseline to evaluate privacy of generative models.

1. Introduction

An ideal generative algorithm should be privacy-preserving and learn the overall data distribution without memorizing any training examples. Synthetic data generated from such models should not reveal if any particular data point was part of the original dataset. In this work, we leverage memorization, which occurs when generated samples cluster around real training points, as a proxy to audit the ϵ-differential privacy (ϵ-DP) guarantees of structured generative and foundation models. We focus on tabular models and frame their synthetic outputs as *approximate reconstructions of training data*. Next, we assess privacy risk by computing the nearest-neighbor distances between real and generated samples.

Our work addresses a central challenge in benchmarking tabular generative and foundation models: how to provide quantifiable privacy guarantees to prove if a model is memorizing. Our auditing technique offers a practical mechanism for *certifiably detecting* when a model has memorized its inputs, thus alerting of potential privacy risks. This assurance can provide a baseline for the privacy assessment of tabular models and encourage their wider adoption.

Audit idea. Synthetic data generated from tabular models are typically evaluated for data fidelity by comparing data distributions and for data utility by checking Machine Learning (ML) performance, respectively. A recent emphasis from global data protection regulations is on provable data protection guarantees. Our auditing addresses this by providing a tight and interpretable lower bound on the ϵ-DP level of tabular models. It can operate without requiring model retraining, adversarial crafting, or access to internal parameters, making it suitable for real-world use.

The core idea behind our approach is rooted in the definition of ϵ-DP: if a mechanism satisfies ϵ-DP, then the posterior distribution over audit inputs given the synthetic outputs must remain close to the prior; assumed here to be uniform over a region $\mathcal{X}$. When generated samples cluster tightly around audit samples, they effectively reveal "hot-spots" where those examples are likely to reside. This posterior deviation violates the ϵ-DP condition, allowing us to confidently reject the hypothesis that the generative mechanism satisfies the claimed privacy guarantee. Our main result formalizes this observation, showing that smaller nearest-neighbor distances imply provably larger lower bounds on the model's true ϵ value, with high statistical confidence.

Procedure. We begin by uniformly sampling m audit examples within a unit-volume region $\mathcal{X} \subset \mathbb{R}^d$ in the training data's d-dimensional space, ideally from an unused area to minimize impact on generative capabilities. The generative model G is then trained on these m audit examples (optionally alongside real training data) and used to generate n synthetic samples, which can optionally be restricted to region $\mathcal{X}$ via rejection sampling for improved audit quality. Finally, we obtain a lower bound on the data pipeline's differential privacy parameter ϵ by computing nearest-neighbor distances between the audit and synthetic samples.

Comparison with existing work. What distinguishes our approach from the existing literature is that *we do not rely on membership inference attacks*, which has a major limita-

[1]National University of Singapore [2]Betterdata.ai. Correspondence to: Rishav Chourasia <rishav1@comp.nus.edu.sg>.

Proceedings of the 1st ICML Workshop on Foundation Models for Structured Data, Vancouver, Canada. 2025. Copyright 2025 by the author(s).

tion when it comes to auditing privacy in machine learning. Each membership prediction on a target tests for a single bit regarding whether the target was included or excluded from the training dataset, which provides very limited information for auditing differential privacy—even if the attack succeeds 100% of the time on m audit targets, the largest lower bound on privacy parameter ϵ derivable that holds with $(1-\beta)100\%$ probability is $\epsilon \geq \log(\sqrt[m]{\beta}/(1-\sqrt[m]{\beta}))$, which is only ≈ 12.71 for $m = 1,000,000$ targets at a confidence level of 95% (i.e. for $\beta = 0.05$). In contrast, *a reconstruction attack seeks to predict all the bits that make up the target*, which can be *infinitely more informative than the outcome of a membership inference attack*. Our main result shows that with m audit examples in d dimensions, observing nearest-neighbor distance sum $\hat{\nu}$ from n reconstructed samples results in a lower bound $\epsilon \geq \log(\Gamma(d/2)/\Gamma(d)) + \log\left(\sqrt[m]{\beta \cdot (md)!}\Big/ 2\pi^{d/2} n \hat{\nu}^d\right)$, that holds with $(1-\beta)100\%$ probability. To contextualize, for just $m = n = 10$ audit and synthetic samples in $d = 10$ dimensions, a nearest-neighbor distance sum $\hat{\nu} \leq 1$ yields $\epsilon \geq 17.34$, $\hat{\nu} \leq 0.1$ yields $\epsilon \geq 40.36$, and $\hat{\nu} \leq 0.01$ yields $\epsilon \geq 63.39$ with 99.9% confidence. For larger and more practical values of audit examples m, synthetic samples n and dimension d, our lower bounds improve substantially.

Relevance. Our auditing method is particularly well-suited for structured generative and foundation models trained on tabular data, where model scale and training cost prohibit multiple training runs. Auditing privacy with membership inference is cumbersome, requiring outlier selection as audit targets, designing a discriminator for membership detection, and performing shadow runs to train the discriminator (Shokri et al., 2017). In contrast, our scheme is remarkably simpler: viewing synthetic samples as reconstruction attempts means auditing involves merely generating samples and computing distances to audit examples. By filling the privacy assessment gap in synthetic data benchmarks, our audit methodology serves as a key enabler for quantifiable privacy evaluations of contemporary generative models.

Related work. The prior literature on privacy auditing schemes is based on conducting membership inference attacks over multiple independent runs of a private algorithm on adjacent datasets that differ in a single record to directly test the ϵ-DP inequality (Gilbert & McMillan, 2018; Ding et al., 2018; Bichsel et al., 2021; 2018; Zanella-Beguelin et al., 2023; Niu et al., 2022). These approaches do not scale for evaluating privacy of large foundational models as even a single training run is costly (Neel & Chang, 2023).

The work of (Steinke et al., 2023) removes the multi-run limitation by noting that the DP guarantee *holds simultaneously* for all the input data points. Their audit strategy exploits the parallelism of including or excluding multiple audit targets in a single run, linking membership-inference accuracy to high-confidence bounds on the algorithm's DP parameters. Our main result exploits not only this parallelism across multiple targets, but also *parallelism across the possible high-dimensional values each target can have*.

Prior audit methods are impractical for generative models due to their reliance on membership inference, retraining, or access to model internals. Our work provides a low-overhead, practically deployable privacy audit for generative model training pipelines that are increasingly treated as black boxes in the industry. As such, our contributions are:

1. A one-run black-box audit strategy for tabular models.
2. A novel view of synthetic samples as reconstructions; using it to audit privacy better than inference attacks.
3. An ϵ-DP audit instantiation that provides a substantially tighter lower bound compared to prior methods.
4. Demonstration on generative models including GANs, VAEs, diffusion, and LLM-based synthesizers.

2. Background

The generation of structured synthetic data from real data typically follows a number of steps. First, the original records from a data universe $\mathcal{D}$, potentially with heterogeneous features, are transformed into corresponding d-dimensional feature vectors via an *invertible transformer* $\mathcal{T} : \mathcal{D} \to \mathbb{R}^d$. For instance, in case of tabular data, numerical features can be normalization or standardization; categorical features are often encoded as one-hot vectors; and ordinal features may be mapped to an interval scale. This transformation yields a vectorized training dataset $D \subset \mathbb{R}^d$.

Next, a generative model G (e.g. GAN (Xu et al., 2019b; Zhao et al., 2022), diffusion model (Kotelnikov et al., 2023), LLM model (Zhao et al., 2023; Wang et al., 2024)) is either trained or fine-tuned on the training dataset D using gradient descent to learn the underlying real distribution. To create synthetic samples in the original format, a set of new d-dimensional vectors $\mathbf{S}$ is sampled from the generative model G, which are then transformed back using $\mathcal{T}^{-1} : \mathbb{R}^d \to \mathcal{D}$.

A formal guarantee is often desired to prevent generative models from memorizing sensitive training data; Differential privacy is the gold standard for quantifying this risk.

Definition 2.1 (Differential Privacy (Dwork et al., 2006b;a))**.** Let $\mathcal{A} : \mathcal{X}^m \to \mathcal{Y}$ be a randomized algorithm. We say $\mathcal{A}$ is ϵ-differentially private (ϵ-DP) if, for all $\mathbf{X}, \mathbf{X}' \in \mathcal{X}^m$ that differ only by replacement of one element, we have

$$\forall S \subset \mathcal{Y}, \quad \Pr[\mathcal{A}(\mathbf{X}) \in S] \leq e^{\epsilon} \cdot \Pr[\mathcal{A}(\mathbf{X}') \in S]. \quad (1)$$

3. Auditing Process

Our audit process evaluates the privacy of the main segment of the real-to-synthetic genAI pipeline that lies between the

Algorithm 1 ϵ-DP Auditor using NN distance sum

Require: Algorithm $\mathcal{A}$ to audit, number of audit examples m, number of synthetic samples n, dimension of feature vectors d, significance level β.

Uniformly sample $\mathbf{X} = (X_1, \cdots, X_m) \xleftarrow{\$} [0,1]^{d\times m}$
Generate synthetic samples $(S_1, \cdots, S_n) \xleftarrow{\$} \mathcal{A}(\mathbf{X})$
Initialize $\hat{\nu} \leftarrow 0$
for $i = 1$ **to** m **do**
  $\hat{\nu} \leftarrow \hat{\nu} + \inf_{j\in[n]} \|X_i - S_j\|_2$
end for
return $\epsilon_{\text{lower}} \leftarrow \max\left\{0, \log\left(\frac{\Gamma(d/2)}{\Gamma(d)} \cdot \frac{\sqrt[m]{\beta\cdot(md)!}}{2\pi^{d/2} n\cdot\hat{\nu}^d}\right)\right\}$

transformation step $\mathcal{T}$ on the input training data and the inverse transformation step $\mathcal{T}^{-1}$ on the output synthetic data. This audit begins by uniformly sampling m vectors in the d-dimensional hypercube $\mathcal{X} = [0,1]^d \subset \mathbb{R}^d$, which we call *audit examples* and denote $\mathbf{X} \in \mathcal{X}^m$. The hypercube $\mathcal{X}$ has a unit volume and can be located anywhere within the d-dimensional space $\mathbb{R}^d$ (where the training data resides), rather than being fixed at the origin 0^d. We then use audit examples, optionally along with the real training dataset $D \subset \mathbb{R}^{n\times d}$, to train a generative model G as usual in the next step of the genAI pipeline. After training, we sample n synthetic feature vectors $\mathbf{S} = (S_1, \cdots, S_n)$ in $\mathbb{R}^d$. If sampling is inexpensive, we can optionally restrict synthetic samples $\mathbf{S}$ to $\mathcal{X}$ via rejection sampling for better audit performance. We denote this audit-to-synthetic generation process as the algorithm $\mathcal{A} : \mathcal{X}^m \to \mathbb{R}^{n\times d}$, where the real training dataset D (if used) is considered fixed and "hardcoded" into $\mathcal{A}$ itself.

In the next phase, we compute the sum of the Euclidean distance $\hat{\nu}(\mathbf{X}, \mathbf{S})$ between the audit examples $\mathbf{X}$ and their nearest neighbors among synthetic examples $\mathbf{S}$, which gives us a proxy measurement of the level of memorization exhibited by the intermediate generative model G in the pipeline.

In the last phase, we convert the measurement $\hat{\nu}$ into an estimate ϵ_{lower} that lower-bounds $\mathcal{A}$'s true differential privacy parameter with probability $(1-\beta)100\%$. This conversion is enabled by the main result of our paper, which is presented in Theorem 3.1. In Algorithm 1, we present the pseudocode of our auditing procedure and also provide a numerically-stable Python implementation in Appendix B.

Theorem 3.1 (Main Result)**.** *Let $\mathcal{A} : [0,1]^{m\times d} \to \mathbb{R}^{n\times d}$ satisfy ϵ-DP. Let $\mathbf{X} = (X_1, \cdots, X_m) \in [0,1]^{m\times d}$ be uniformly random. Let $\mathbf{S} = (S_1, \cdots, S_n) = \mathcal{A}(\mathbf{X})$. Let $\mathbf{d}^{(1)}_{X_i} = \inf_{j\in[n]} \|X_i - S_j\|_2$. Then, for all $\nu \geq 0$ and $s \in \mathbb{R}^n$ in the support of $\mathbf{S}$,*

$$\Pr\left[\sum_{i=1}^{m} \mathbf{d}^{(1)}_{X_i} \leq \nu \,\middle|\, \mathbf{S} = s\right] \leq \frac{1}{(md)!}\left(2\pi^{\frac{d}{2}} n e^{\epsilon} \nu^d \frac{\Gamma(d)}{\Gamma(\frac{d}{2})}\right)^m.$$

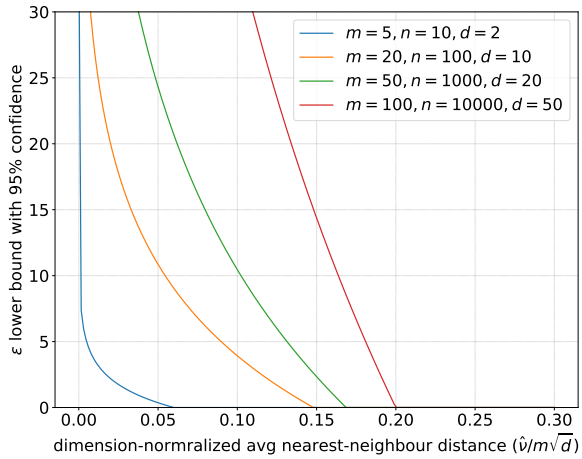

Figure 1. Lower bound on the DP parameter ϵ from Theorem 3.1 as the memorization metric ($\hat{\nu}$) changes for different choices of audit examples m, synthetic samples n, and dimension d.

Theorem 3.1 gives a way to test the hypothesis that "algorithm $\mathcal{A}$ is ϵ-DP". This can be seen by rearranging the terms in the equation and taking expectation over the distribution of $\mathbf{S}$ to get the following statement: If $\mathcal{A}$ is ϵ-DP, then

$$\Pr_{\substack{\mathbf{X}\leftarrow[0,1]^d \\ \mathbf{S}\leftarrow\mathcal{A}(\mathbf{X})}}\left[\hat{\nu}(\mathbf{X}, \mathbf{S}) \leq \left(\frac{\Gamma(d/2)}{\Gamma(d)} \cdot \frac{\sqrt[m]{\beta\cdot(md)!}}{2\pi^{d/2} n e^{\epsilon}}\right)^{\frac{1}{d}}\right] \leq \beta,$$

where $\hat{\nu}(\mathbf{X}, \mathbf{S}) := \sum_{i=1}^{m} \inf_{j\in[n]} \|X_i - S_j\|_2$ is the sum of the nearest-neighbour (NN) distances of audit examples to synthetic samples. Put differently, under the assumption that $\mathcal{A}$ is ϵ-DP, the metric $\hat{\nu}(\mathbf{X}, \mathbf{S})$ will exceed the constant $\nu(\epsilon, \beta, m, n, d)$ with high probability. Thus, if the observed value of $\hat{\nu}(\mathbf{X}, \mathbf{S})$ is less than ν, we can confidently reject the ϵ-DP claim for $\mathcal{A}$, with a false rejection probability $\leq \beta$.

Our auditor in Algorithm 1 uses the hypothesis test for ϵ-DP in Theorem 3.1 to estimate a lower bound ϵ_{lower} on the true ϵ-DP parameter of $\mathcal{A}$, facilitated by the following lemma.

Lemma 3.2 ((Steinke et al., 2023))**.** *For each $\mathcal{A}$, let $\hat{\nu}_{\mathcal{A}} \in \Omega$ be a random variable and let $P_{\mathcal{A}} \in \mathbb{R}^+$ be a fixed number. For each $\epsilon, \beta > 0$, let $T_{\epsilon,\beta} \subset \Omega$ satisfy*

$$\forall \mathcal{A} \;:\; (P_{\mathcal{A}} = \epsilon \quad \Longrightarrow \quad \Pr[\hat{\nu}_{\mathcal{A}} \in T_{\epsilon,\beta}] \leq \beta).$$

Further suppose that, if $\epsilon_1 \leq \epsilon_2$, then $T_{\epsilon_1,\beta} \supset T_{\epsilon_2,\beta}$. Then, for all $\mathcal{A}$ and all $\beta > 0$,

$$\Pr[P_{\mathcal{A}} \geq \sup\{\epsilon > 0 \;:\; \hat{\nu}_{\mathcal{A}} \in T_{\epsilon,\beta}\}] \geq 1 - \beta.$$

The idea behind this conversion is as follows. Suppose, for all choices of $\epsilon, \beta > 0$, we can identify a set $T_{\epsilon,\beta}$ of extreme values for a statistic $\hat{\nu}_{\mathcal{A}} \in \Omega$ that rarely occurs (with probability $\leq \beta$) whenever the null hypothesis "$\mathcal{A}$ is ϵ-DP" is true ($P_{\mathcal{A}} = \epsilon$). Then given an observation of the statistic $\hat{\nu}$, we can reject the null hypothesis for all values

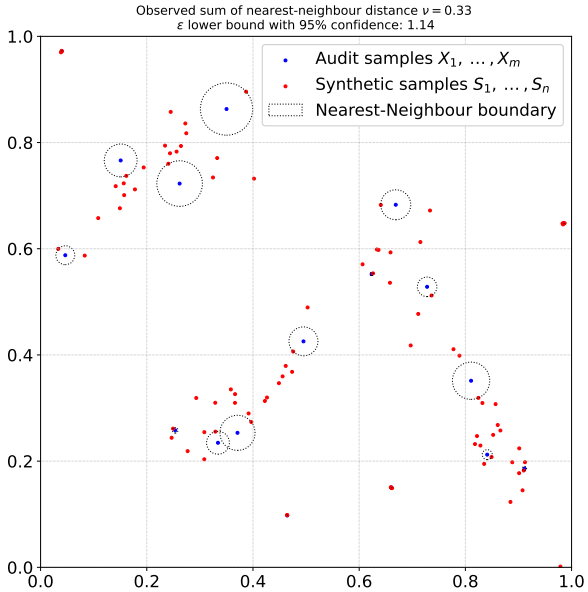

Figure 2. Our ϵ-DP audit visualization in 2D. Audit dataset $\mathbf{X}$ consists of $m = 20$ uniformly random points in unit square. We train a Gaussian Mixture model (8 components) on $\mathbf{X}$ and generate synthetic dataset $\mathbf{S}$ containing $n = 50$ samples.

of ϵ for which the set $T_{\epsilon,\beta}$ contains $\hat{\nu}$ and be rarely wrong (with probability $\leq \beta$). The maximal such ϵ_{lower} that we can reject is essentially a lower bound estimate of the true privacy parameter $P_{\mathcal{A}}$ of the algorithm $\mathcal{A}$ that holds with high probability (i.e., with probability $\geq 1 - \beta$). Figure 1 plots the lower bound ϵ_{lower} as a function of normalized $\hat{\nu}$ and Figure 3 in the appendix visualizes the p-value (β) of rejecting ϵ-DP hypothesis given an observation value $\hat{\nu}$.

4. Experiments

In this section, we present results from applying our ϵ-DP audit scheme to several synthetic data generation algorithms from the literature. While our scheme can audit general structured generative AI, we focus our experiments on tabular data synthesis techniques to align with the workshop.

Toy experiment. Figure 2 shows the privacy audit of an 8-component Scikit-Learn Gaussian mixture model. This model was trained on a tiny dataset of $m = 20$ random 2D samples and used to generate $n = 50$ synthetic samples, which resulted in $\epsilon_{\text{lower}} = 1.14$ with 95%-confidence.

Large-scale experiments. To demonstrate the effectiveness of our auditing scheme, we select six state-of-the-art tabular generative models to evaluate: CTGAN (Xu et al., 2019a), TVAE (Xu et al., 2019a), CTAB-GAN+ (Zhao et al., 2022), ARF (Watson et al., 2023), Tabula (Zhao et al., 2023) and TabDiff (Shi et al., 2024). These algorithms cover a wide range of generative models, including GANs, VAEs, tree-based models, LLMs, and diffusion models. For consistent

Table 1. DP audit results for various synthetic data generation approaches in literature (holds with 95%-confidence).

ALGORITHM	ϵ_{lower}
GAUSSIAN-MIXTURE (CF. FIGURE 2)	1.14
CTGAN (XU ET AL., 2019A)	0
CTAB-GAN+ (ZHAO ET AL., 2022)	0.04
TVAE (XU ET AL., 2019A)	4.42
ARF (WATSON ET AL., 2023)	2.26
TABDIFF (SHI ET AL., 2024)	2.36
TABULA (ZHAO ET AL., 2023)	57.53

auditing across all evaluations (except the toy example), we uniformly fix the parameters at $m = 100$, $n = 1,000$, and $d = 60$. These settings reflect typical production environments: $d \leq 60$ covers most tabular datasets encountered, $m = 100$ audit examples represent a negligible fraction ($< 2\%$) of typical training dataset sizes, and sampling $n = 1,000$ synthetic data is relatively inexpensive.

To achieve the best audit results in these experiments, we used only the audit dataset $\mathbf{X} \in \mathcal{X}^m$, omitting a separate real training dataset D and restrict the synthetic samples to $\mathbf{S} \in \mathcal{X}^n$ (a configuration that our scheme supports). Given this small effective training size ($m = 100$), we increased the training duration for the evaluated models: CTGAN, TVAE, and CTAB-GAN+ were trained for 2000 epochs, Tabula for 1000, and TabDiff for 8000 iterations. Table 1 summarizes the results of our privacy auditing.

5. Conclusion and Future Work

We presented a novel and efficient black-box technique for auditing the differential privacy level of structured synthetic data generators. By reframing the problem around the concept of training data reconstruction and utilizing nearest-neighbor distances between training and synthetic data as a sensitive indicator of memorization, our approach provides substantially larger lower bounds on the privacy parameter ϵ compared to other audit techniques based on membership inference attacks. This one-run auditing strategy is efficient, avoids the need for adversarial crafting or calibration via shadow models, and its minimal assumptions make it widely applicable for assessing the privacy guarantees of various synthetic data generation systems in practice. Our work directly supports the goal of building trust and accountability in structured generative and foundation models by offering a principled, scalable, and interpretable privacy auditing tool tailored for real-world use. In doing so, it contributes to a broader goal of providing a framework for quantifiable privacy guarantees for synthetic data, critical for evaluating the current tabular models and building the future ones.

Future work. An important area for future research involves adapting our audit approach to other notions of differential privacy, including (ϵ, δ)-DP, f-DP, and Rényi DP.

References

Bichsel, B., Gehr, T., Drachsler-Cohen, D., Tsankov, P., and Vechev, M. Dp-finder: Finding differential privacy violations by sampling and optimization. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pp. 508–524, 2018.

Bichsel, B., Steffen, S., Bogunovic, I., and Vechev, M. Dp-sniper: Black-box discovery of differential privacy violations using classifiers. In *2021 IEEE Symposium on Security and Privacy (SP)*, pp. 391–409. IEEE, 2021.

Ding, Z., Wang, Y., Wang, G., Zhang, D., and Kifer, D. Detecting violations of differential privacy. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pp. 475–489, 2018.

Dwork, C., Kenthapadi, K., McSherry, F., Mironov, I., and Naor, M. Our data, ourselves: Privacy via distributed noise generation. In *Advances in cryptology-EUROCRYPT 2006: 24th annual international conference on the theory and applications of cryptographic techniques, st. Petersburg, Russia, May 28-June 1, 2006. proceedings 25*, pp. 486–503. Springer, 2006a.

Dwork, C., McSherry, F., Nissim, K., and Smith, A. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography: Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006. Proceedings 3*, pp. 265–284. Springer, 2006b.

Gilbert, A. C. and McMillan, A. Property testing for differential privacy. In *2018 56th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pp. 249–258. IEEE, 2018.

Kotelnikov, A., Baranchuk, D., Rubachev, I., and Babenko, A. Tabddpm: Modelling tabular data with diffusion models. In *International Conference on Machine Learning*, pp. 17564–17579. PMLR, 2023.

Neel, S. and Chang, P. Privacy issues in large language models: A survey. *arXiv preprint:2312.06717*, 2023.

Niu, B., Zhou, Z., Chen, Y., Cao, J., and Li, F. Dp-opt: identify high differential privacy violation by optimization. In *International Conference on Wireless Algorithms, Systems, and Applications*, pp. 406–416. Springer, 2022.

Shi, J., Xu, M., Hua, H., Zhang, H., Ermon, S., and Leskovec, J. Tabdiff: a unified diffusion model for multimodal tabular data generation. In *NeurIPS 2024 Third Table Representation Learning Workshop*, 2024.

Shokri, R., Stronati, M., Song, C., and Shmatikov, V. Membership inference attacks against machine learning models. In *2017 IEEE symposium on security and privacy (SP)*, pp. 3–18. IEEE, 2017.

Steinke, T., Nasr, M., and Jagielski, M. Privacy auditing with one (1) training run. *Advances in Neural Information Processing Systems*, 36:49268–49280, 2023.

Wang, Y., Feng, D., Dai, Y., Chen, Z., Huang, J., Ananiadou, S., Xie, Q., and Wang, H. Harmonic: Harnessing llms for tabular data synthesis and privacy protection. *arXiv preprint:2408.02927*, 2024.

Watson, D. S., Blesch, K., Kapar, J., and Wright, M. N. Adversarial random forests for density estimation and generative modeling. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 2023.

Xu, L., Skoularidou, M., Cuesta-Infante, A., and Veeramachaneni, K. Modeling tabular data using conditional gan. In *Advances in Neural Information Processing Systems, 2019*, volume 32, pp. 7335–7345. Curran Associates, Inc., 2019a.

Xu, L., Skoularidou, M., Cuesta-Infante, A., and Veeramachaneni, K. Modeling tabular data using conditional gan. *Advances in neural information processing systems*, 32, 2019b.

Zanella-Beguelin, S., Wutschitz, L., Tople, S., Salem, A., Rühle, V., Paverd, A., Naseri, M., Köpf, B., and Jones, D. Bayesian estimation of differential privacy. In *International Conference on Machine Learning*, pp. 40624–40636. PMLR, 2023.

Zhao, Z., Kunar, A., Birke, R., and Chen, L. Y. Ctab-gan+: Enhancing tabular data synthesis. *arXiv preprint:2204.00401*, 2022.

Zhao, Z., Birke, R., and Chen, L. Tabula: Harnessing language models for tabular data synthesis. *arXiv preprint:2310.12746*, 2023.

A. Deferred Proofs

Lemma A.1 ((Steinke et al., 2023)). *Suppose for all $\nu \in \mathbb{R}$ we have $\Pr[X_1 \le \nu] \le \Pr[Y_1 \le \nu]$ and for all $x, \nu \in \mathbb{R}$ we have $\Pr[X_2 \le \nu | X_1 = x] \le \Pr[Y_2 \le \nu]$. Assume that Y_1 and Y_2 are independent. Then for all $\nu \in \mathbb{R}$ we have $\Pr[X_1 + X_2] \le \Pr[Y_1 + Y_2]$.*

Proof. For all $\nu \in \mathbb{R}$, we have

$$\begin{aligned}
\Pr[X_1 + X_2 \le \nu] &= \mathop{\mathbb{E}}_{X_1}\left[\Pr_{X_2}[X_2 \le \nu - X_1 | X_1]\right] \\
&\le \mathop{\mathbb{E}}_{X_1}\left[\Pr_{Y_2}[Y_2 \le \nu - X_1]\right] \\
&= \mathop{\mathbb{E}}_{Y_2}\left[\Pr_{X_1}[X_1 \le \nu - Y_2]\right] \\
&\le \mathop{\mathbb{E}}_{Y_2}\left[\Pr_{Y_1}[Y_1 \le \nu - Y_2]\right] \\
&= \Pr[Y_1 + Y_2 \le \nu].
\end{aligned}$$

□

Theorem 3.1 (Main Result). *Let $\mathcal{A} : [0,1]^{m \times d} \to \mathbb{R}^{n \times d}$ satisfy ϵ-DP. Let $\mathbf{X} = (X_1, \cdots, X_m) \in [0,1]^{m \times d}$ be uniformly random. Let $\mathbf{S} = (S_1, \cdots, S_n) = \mathcal{A}(\mathbf{X})$. Let $\mathbf{d}^{(1)}_{X_i} = \inf_{j \in [n]} \|X_i - S_j\|_2$. Then, for all $\nu \ge 0$ and $s \in \mathbb{R}^n$ in the support of $\mathbf{S}$,*

$$\Pr\left[\sum_{i=1}^{m} \mathbf{d}^{(1)}_{X_i} \le \nu \middle| \mathbf{S} = s\right] \le \frac{1}{(md)!}\left(2\pi^{\frac{d}{2}} n e^{\epsilon} \nu^d \frac{\Gamma(d)}{\Gamma(\frac{d}{2})}\right)^m.$$

Proof. Assume M is ϵ-DP. Fix some synthetic data $s \in \mathbb{R}^n$. We now analyze the distribution of input $\mathbf{X}$ conditioned on $\mathbf{S} = s$. Fix some $i \in [n]$ and $x_{<i} \in \mathcal{X}^{i-1}$. By Bayes law,

$$\begin{aligned}
p(X_i = x | \mathbf{S} = s, \mathbf{X}_{<i} = x_{<i}) &= \frac{p(\mathbf{S} = s | X_i = x, \mathbf{X}_{<i} = x_{<i})}{p(\mathbf{S} = s | \mathbf{X}_{<i} = x_{<i})} \times p(X_i = x | \mathbf{X}_{<i} = x_{<i}) \\
&= \frac{p(X_i = x | \mathbf{X}_{<i} = x_{<i})}{\int_{\mathcal{X}} \frac{p(\mathbf{S}=s | X_i = x', \mathbf{X}_{<i} = x_{<i})}{p(\mathbf{S}=s | X_i = x, \mathbf{X}_{<i} = x_{<i})} p(X_i = x' | \mathbf{X}_{<i} = x_{<i}) \mathrm{d}x'} \\
&= \frac{p(X_i = x)}{\int_{\mathcal{X}} \frac{p(\mathbf{S}=s | X_i = x', \mathbf{X}_{<i} = x_{<i})}{p(\mathbf{S}=s | X_i = x, \mathbf{X}_{<i} = x_{<i})} p(X_i = x') \mathrm{d}x'} && \text{(Since } X_i\text{s are independent)} \\
&= \frac{1}{\int_{\mathcal{X}} \frac{p(\mathbf{S}=s | X_i = x', \mathbf{X}_{<i} = x_{<i})}{p(\mathbf{S}=s | X_i = x, \mathbf{X}_{<i} = x_{<i})} \mathrm{d}x'} && \text{(Since } p(X_i = x) = 1 \text{ for all } x \in [0,1]^d) \\
&\in [e^{-\epsilon}, e^{\epsilon}] && \text{(Since } \mathcal{A} \text{ is } \epsilon\text{-DP)}
\end{aligned}$$

Basically, this says that the posterior distribution of X_i given the observed synthetic data $\mathbf{S}$ (and previous audit examples) is still ϵ-close to being uniformly distributed on the hypercube $[0,1]^d$.

We are interested in the random variables $\mathbf{d}^{(1)}_{X_1}, \cdots, \mathbf{d}^{(1)}_{X_m}$ which denotes the euclidean distance between $X_1, \cdots, X_m$ and their respective nearest neighbors among $S_1, \cdots, S_n$. But these random variables are *dependent* on one another. So, our goal at hand is to come up with random variables $Y_1, \cdots, Y_m \in \mathcal{X}$ that are mutually independent when conditioned on the synthetic data $\mathbf{S} = s$ such that $\mathbf{d}^{(1)}_{Y_i}$ dominates $\mathbf{d}^{(1)}_{X_i}$ as follows:

$$\forall i \in [m] \ : \ \Pr\left[\mathbf{d}^{(1)}_{X_i} \le \nu \middle| \mathbf{S} = s, \mathbf{X}_{<i} = x_{<i}\right] \le \Pr\left[\mathbf{d}^{(1)}_{Y_i} \le \nu \middle| \mathbf{S} = s\right]$$

If we could come up with a construction for such new random variables $Y_1, \cdots, Y_n$, then it would immediately follow from Lemma A.1 that

$$\Pr\left[\sum_{i=1}^{m} \mathbf{d}^{(1)}_{X_i} \le \nu \middle| \mathbf{S} = s\right] \le \Pr\left[\sum_{i=1}^{m} \mathbf{d}^{(1)}_{Y_i} \le \nu \middle| \mathbf{S} = s\right]$$

And since $\mathbf{d}^{(1)}_{Y_1}, \cdots, \mathbf{d}^{(1)}_{Y_m}$ will also be independent conditioned on $\mathbf{S} = s$, we can easily bound the RHS.

Question. What is the Y_i random variables we trying to construct?

We are given n points $s_1, \cdots, s_n \in \mathbb{R}^d$ within it. We want to construct a distribution p^* on the hypercube that is ϵ-close to the uniform distribution (i.e. $p^*(x) \in [e^{-\epsilon}, e^{\epsilon}]$ for all x) such that the distance $\mathbf{d}_{Y_i}$ of a $Y_i \sim p^*$ from the nearest neighbor among $s_1, \cdots, s_n$ is likely to be smaller than among all other distributions p that are also ϵ-close to uniform:

$$\forall 0 \leq \nu \ : \Pr_{Y_i \sim p}\left[\mathbf{d}^{(1)}_{Y_i} \leq \nu \middle| \mathbf{S} = s\right] \leq \Pr_{Y_i \sim p^*}\left[\mathbf{d}^{(1)}_{Y_i} \leq \nu \middle| \mathbf{S} = s\right].$$

Since the random variable $\mathbf{d}^{(1)}_{X_i}$ given $\mathbf{S} = s$ and $\mathbf{X}_{<i} = x_{<i}$ is also ϵ-close to being uniform, the random variable $\mathbf{d}^{(1)}_{Y_i}$ where $Y_i \sim p^*$ will dominate $\mathbf{d}^{(1)}_{X_i}$ *by construction*.

From geometry, this distribution p^* must give the higher density e^{ϵ} to all the n balls $B_r(s_1), \cdots, B_r(s_n)$ that intersect with the unit hypercube $[0,1]^d$ and $e^{-\epsilon}$ everywhere else inside the hypercube. The value of radius r so that p^* is a valid distribution can be computed geometrically, but our analysis doesn't need the exact value of r (although our analysis can be improved by computing limits on this r).

For $Y_i \sim p^*$ we can find an upper bound on the density of $\mathbf{d}^{(1)}_{Y_i}$ as follows

$$\begin{aligned}
p\left(\mathbf{d}^{(1)}_{Y_i} = \nu \middle| \mathbf{S} = s\right) &= \int_{\{y \in [0,1]^d : \exists i \in [n] \text{ s.t. } d(y, s_i) = \nu\}} p(Y_i = y | \mathbf{S} = s) \mathrm{d}y \\
&\leq \int_{\{y \in [0,1]^d : \exists i \in [n] \text{ s.t. } d(y, s_i) = \nu\}} e^{\epsilon} \mathrm{d}y && \text{(From } \epsilon\text{-closeness to uniform)} \\
&\leq \sum_{i=1}^{n} e^{\epsilon} \cdot \int_{\{y \in [0,1]^d : d(y, s_i) = \nu\}} \mathrm{d}y \\
&\leq \sum_{i=1}^{n} e^{\epsilon} \cdot \int_{B_\nu(s_i)} \mathrm{d}y \\
&= \sum_{i=1}^{n} e^{\epsilon} \cdot A_{d-1}(\nu) && \text{(where } A_{d-1}(\nu) \text{ is the surface area of } B_\nu(s_i)) \\
&= \underbrace{n e^{\epsilon} \cdot \frac{2\pi^{d/2}}{\Gamma(\frac{d}{2})}}_{\text{Let constant } a =} \cdot \nu^{d-1}.
\end{aligned}$$

Since $\mathbf{d}^{(1)}_{Y_1}, \cdots, \mathbf{d}^{(1)}_{Y_m}$ are conditionally independent, the conditional density of the sum $\sum_{i=1}^{m} \mathbf{d}^{(1)}_{Y_i}$ is the convolution of conditional densities of $\mathbf{d}^{(1)}_{Y_1}, \cdots, \mathbf{d}^{(1)}_{Y_m}$, which is upper bounded by the m-fold convolution of $f_d(\nu) := a \cdot \nu^{d-1}$, as follows.

$$p\left(\sum_{i=1}^{m} \mathbf{d}^{(1)}_{Y_i} = \nu \middle| \mathbf{S} = s\right) \leq f_d^{\circledast m}(\nu)$$

Let's solve the m-fold convolution of $f_d(\nu) = a \cdot \nu^{d-1}$. Laplace transform[1] of the function f_d is

$$\mathcal{L}\{f_d(t)\}(s) = a \cdot \mathcal{L}\{t^{d-1}\}(s) = a \cdot \frac{\Gamma(d)}{s^d}.$$

From the convolution property of Laplace transforms

$$\mathcal{L}\{f^{\circledast m}(t)\}(s) = a^m \cdot \prod_{i=1}^{m} \mathcal{L}\{f(t)\}(s) = a^m \cdot \frac{\Gamma(d)^m}{s^{md}} = a^m \cdot \frac{\Gamma(d)^m}{\Gamma(md)} \mathcal{L}\{t^{md-1}\}(s) = \mathcal{L}\left\{a^m \cdot \frac{\Gamma(d)^m}{\Gamma(md)} \cdot t^{md-1}\right\}(s)$$

[1] Laplace transform of function f is defined as $\mathcal{L}\{f\}(s) := \int_0^\infty e^{-st} \cdot f(t) \mathrm{d}t$. For any two functions f, g, the Laplace transform of the convolution $(f \circledast g)(t) = \int_0^t f(t-\tau) \cdot g(\tau) \mathrm{d}\tau$ is the product of the respective Laplace transforms: $\mathcal{L}\{f \circledast g\} = \mathcal{L}\{f\} \cdot \mathcal{L}\{g\}$.

Therefore,

$$f^{\circledast m}(t) = a^m \cdot \frac{\Gamma(d)^m}{\Gamma(md)} \cdot t^{md-1}.$$

This means,

$$p\left(\sum_{i=1}^{m} \mathbf{d}_{Y_i}^{(1)} = \nu \middle| \mathbf{S} = s\right) \leq \underbrace{a^m \cdot \frac{\Gamma(d)^m}{\Gamma(md)}}_{\text{Let constant } b=} \cdot \nu^{md-1}.$$

So, probability $\Pr\left[\sum_{i=1}^{m} \mathbf{d}_{Y_i}^{(1)} \leq \nu \middle| \mathbf{S} = s\right]$ can be found by integrating:

$$\Pr\left[\sum_{i=1}^{m} \mathbf{d}_{Y_i}^{(1)} \leq \nu \middle| \mathbf{S} = s\right] \leq b \int_0^{\nu} t^{md-1} \mathrm{d}t = \frac{b}{md} \cdot \nu^{md}.$$

Finally from Lemma A.1, we get

$$\Pr\left[\sum_{i=1}^{m} \mathbf{d}_{X_i}^{(1)} \leq \nu \middle| \mathbf{S} = s\right] \leq \Pr\left[\sum_{i=1}^{m} \mathbf{d}_{Y_i}^{(1)} \leq \nu \middle| \mathbf{S} = s\right] \leq \frac{1}{(md)!} \cdot \left(2\pi^{d/2} n e^{\epsilon} \cdot \frac{\Gamma(d)}{\Gamma(\frac{d}{2})} \cdot \nu^d\right)^m.$$

□

B. Implementation of Our DP Auditor from Theorem 3.1

```python
import numpy as np
from scipy.special import gammaln

# m = number of audit example vectors
# n = number of synthetic sample vectors
# d = dimension of audit and synthetic feature vectors
# v = sum of Euclidean distances between audit examples and NN synthetic samples
# eps = DP guarantee of null hypothesis
# output: p-value = probability of <=v NN distance sum under null hypothesis
def get_pvalue(m, n, d, v, eps):
    assert v > 0
    assert eps >= 0
    log_gamma_term = gammaln(d) - gammaln(d / 2)
    log_md_factorial = gammaln(m * d + 1)
    log_base_terms = np.log(2) + (d / 2) * np.log(np.pi) + np.log(n) + log_gamma_term
    log_p_value = -log_md_factorial + m * (log_base_terms + eps + d * np.log(v))
    p_value = np.exp(log_p_value)
    return np.minimum(p_value, 1)

# m = number of audit example vectors
# n = number of synthetic sample vectors
# d = dimension of audit and synthetic feature vectors
# v = sum of Euclidean distances between audit examples and NN synthetic samples
# p = 1-confidence e.g. p=0.05 corresponds to 95%
# output: lower bound on eps i.e. algorithm is not eps-DP
def get_epslb(m, n, d, v, p):
    assert v > 0
    assert p > 0
    log_gamma_term = gammaln(d/2) - gammaln(d)
    log_md_factorial = gammaln(m * d + 1)
    log_top_terms = (np.log(p) + log_md_factorial) / m
    log_bottom_terms = np.log(2) + (d / 2) * np.log(np.pi) + np.log(n) + d * np.log(v)
    eps_lower = log_gamma_term + log_top_terms - log_bottom_terms
    return np.maximum(0,eps_lower)
```

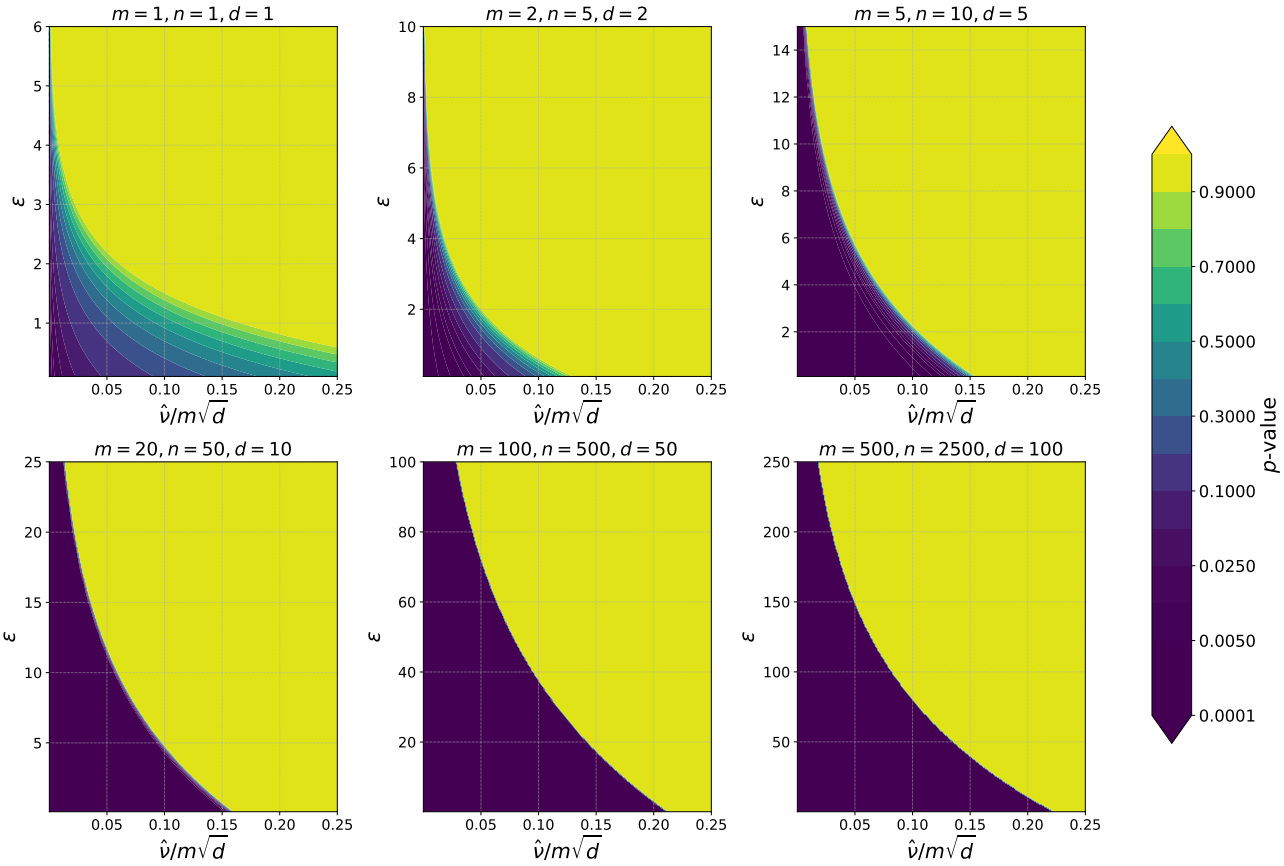

Figure 3. Visualization of Theorem 3.1's p-value, presented using the colormap, for rejecting the null hypothesis "algorithm $\mathcal{A}$ for training a generative model is ϵ-differentially private". Here, m represents the number of uniformly-random audit examples added to the training dataset, n the number of synthetic samples generated from a model trained using $\mathcal{A}$, and d the dimension of training/synthetic datasets. The plot shows how the p-value changes as the memorization metric (which we define as the average distance between audit examples and their respective nearest synthetic samples $\hat{\nu}/m$, normalized by dimension size $\sqrt{d}$) varies across different audit settings.