WHEN TABLES LEAK: ATTACKING DIGIT MEMORIZATION IN LLM-BASED TABULAR DATA GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Large Language Models (LLMs) have recently demonstrated remarkable performance in generating high-quality tabular synthetic data. In practice, two primary approaches have emerged for adapting LLMs to tabular data generation: (i) fine-tuning smaller models directly on tabular datasets, and (ii) prompting larger models with examples provided in context. In this work, we show that popular implementations from both regimes exhibit a tendency to compromise privacy by reproducing memorized patterns of numeric digits from their training data. To systematically analyze this risk, we introduce a simple No-box Membership Inference Attack (MIA) called LevAtt that assumes adversarial access to only the generated synthetic data and targets the string sequences of numeric digits in synthetic observations. Using this approach, our attack exposes substantial privacy leakage across a wide range of models and datasets, and in some cases, is even a perfect membership classifier on state-of-the-art models. Our findings highlight a unique privacy vulnerability of LLM-based synthetic data generation and the need for effective defenses. To this end, we propose two methods, including a novel sampling strategy that strategically perturbs digits during generation. Our evaluation demonstrates that this approach can defeat these attacks with minimal loss to the fidelity and utility of the synthetic data. [1]

1 INTRODUCTION

Machine learning systems across diverse domains—from healthcare databases to financial risk assessment platforms—rely heavily on structured tabular datasets (Giuffré & Shung, 2023; Assefa et al., 2021; Flanagan et al., 2022). This widespread dependence has driven significant interest in synthetic tabular data generation, where computational models learn to produce artificial records that statistically mirror the patterns of real datasets while avoiding direct replication (Fonseca & Bação, 2023). The utility of synthetic data stems from two primary advantages: it can supplement limited training samples to improve model performance on underrepresented populations or rare events, and it facilitates private data sharing by generating records that do not directly correspond to actual individuals. These benefits are especially valuable in privacy-sensitive sectors such as medicine (Vallevik et al., 2024) and banking (Wu et al., 2023), where regulatory constraints often limit access to original data. As a result, synthetic data generation has emerged as an essential technique for expanding machine learning capabilities in data-constrained and privacy-regulated environments(Platzer & Reutterer, 2021; McKenna et al., 2022).

Large Language Models (LLMs) have recently emerged as state-of-the-art tabular synthetic data generators. Unlike traditional approaches—such as Generative Adversarial Networks (GANs), Variational Autoencoders (VAEs), and diffusion models—that operate directly over original feature space, LLMs encode tabular rows into string representations and leverage two primary methodologies for generation. In-context learning (ICL) approaches include existing tabular rows within the LLM's context window and prompt the LLM to generate additional rows following the observed patterns (Seedat et al., 2024; Kim et al., 2024; Hollmann et al., 2025b; Ma et al., 2024), while supervised fine-tuning (SFT) methods train LLMs on larger quantities of string-encoded tabular data to learn the underlying distribution before sampling synthetic records (Borisov et al., 2023; Wang et al., 2025). Both approaches generate synthetic data by producing new string sequences that are subsequently decoded back into tabular format. In both cases, LLM-based generators have been shown to produce synthetic data that exhibits superior statistical fidelity to original datasets and maintains high utility for downstream machine learning tasks. Moreover, initial evaluations suggest these methods may offer enhanced privacy protection relative to conventional approaches (Solatorio & Dupriez, 2023).

However, these architectural differences raise unresolved questions about privacy. Extensive research has demonstrated that LLMs exhibit tendencies to memorize training data, particularly when exposed to repeated

[1] An anonymous code repository is available here.

patterns (Carlini et al., 2021b; Kandpal et al., 2022), longer sequences (Carlini et al., 2023; Wang et al., 2024), or through supervised fine-tuning processes (Chu et al., 2025). These memorization behaviors, which can be beneficial for language modeling tasks, present unique privacy risks in the context of tabular data generation where training data often contain structured, long sequences of repeated values across observations that have little semantic meaning.

In order to measure privacy risk, Membership Inference Attacks (MIAs) (Shokri et al., 2017; Carlini et al., 2021a) have emerged as the linchpin of privacy auditing for tabular synthetic data generators, serving as the primary tool for evaluating privacy risks across diverse generative approaches (Chen et al., 2020; Hilprecht et al., 2019; van Breugel et al., 2023; Ward et al., 2024; 2025b). However, these methods focus exclusively on the feature space over which traditional generative models operate, potentially missing the string-space vulnerabilities introduced by LLM-based generation entirely.

To bridge this gap, we examine privacy risks in LLM-based tabular data generation by introducing LevAtt, an MIA that exploits Levenshtein Distance on the string representations of tabular data—the actual format LLMs generate—rather than the feature space alone. We find through extensive experimentation in both the ICL and SFT regimes that state-of-the-art LLMs often memorize and replicate numeric values from training data, exposing sensitive information digit-for-digit in synthetic outputs. Even under a conservative no-box threat model, where we assume only adversarial access to the synthetic data, we uncover that LLMs can leak private data through memorized digit patterns, revealing vulnerabilities that conventional feature-space MIAs fail to detect.

To address this new-found risk, we study two defenses: an ad-hoc post-processing algorithm we call Digit Modifier (DM) that flips digits to break sequential patterns and a novel Tendency-based Logit Processor (TLP) that strategically perturbs digits at sample time. We find that both strategies can defeat these attacks, however TLP can effectively control for privacy leakage with minimal effect on the statistical fidelity or downstream machine learning utility of the synthetic data.

2 Related Work

2.1 Tabular Data Generation

Tabular generative models aim to learn a generator G from training data T that approximates the true data distribution $p_X(X)$. We represent tabular data as a matrix $\mathbf{X} \in \mathcal{X}^{n \times d}$, where n denotes the number of samples, d the number of features, and $\mathcal{X}$ is the feature value domain. Each row $\mathbf{x}_i \in \mathcal{X}^d$ represents a data point drawn from the underlying distribution $p_X(X)$, while columns correspond to features that may have heterogeneous data types. We denote the j-th feature value of the i-th sample as $\mathbf{x}_{i,j}$. The training dataset $T = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ comprises n independent samples from $p_X(X)$. The learned model generates synthetic samples $\tilde{\mathbf{x}} \sim G$, forming a synthetic dataset $S = \{\tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_2, \ldots, \tilde{\mathbf{x}}_m\}$. In this work, we assume that features can be of a continuous, ordinal, or categorical type.

Deep Learning-Based Generation. In recent years, a variety of conventional tabular synthetic data generators have been proposed including Generative Adversarial Networks (Yoon et al., 2019; 2020; Xu et al., 2019), likelihood-based methods Ankan & Panda (2015); Durkan et al. (2019); Watson et al. (2023), and diffusion models Kotelnikov et al. (2022); Suh et al. (2023); Zhang et al. (2024) Each of these operate directly over the feature space $\mathcal{X}^d$, learning mappings $\mathcal{G} : \mathcal{Z} \to \mathcal{X}^d$ that model the joint distribution $p_X(X)$. Crucially, these approaches fundamentally treat each sample as an atomic unit, generating a complete feature vector $\mathbf{x} \in \mathcal{X}^d$ simultaneously.

LLM-Based Generation In contrast, LLM-based approaches reframe tabular generation as autoregressive text generation. Training samples are encoded into strings, tokenized into sequences from vocabulary $\mathcal{W}$, and the LLM generates according to $p(t) = \prod_{k=1}^{j} p(w_k | w_1, \ldots, w_{k-1})$. Rather than modeling the joint distribution $p_X(X)$ directly, this approach decomposes it through sequential conditioning, generating feature values $x_{i,j}$ token by token based on previously generated values. This sequential generation process fundamentally breaks the atomic unit assumption, introducing new dynamics where generated tokens influence later ones through the autoregressive chain.

Within this paradigm, two complementary generation strategies have emerged based on data availability and computational resources. In-context-based methods leverage large foundation models by presenting tabular examples directly within the context window, enabling few-shot generation without parameter updates in low-data regimes (Seedat et al., 2024; Kim et al., 2024; Hollmann et al., 2025b; Ma et al., 2024). When larger datasets are available, SFT-based methods instead fine-tune smaller language models through direct optimization on tabular generation tasks (Borisov et al., 2023; Solatorio & Dupriez, 2023; Wang et al., 2025). Both strategies main-

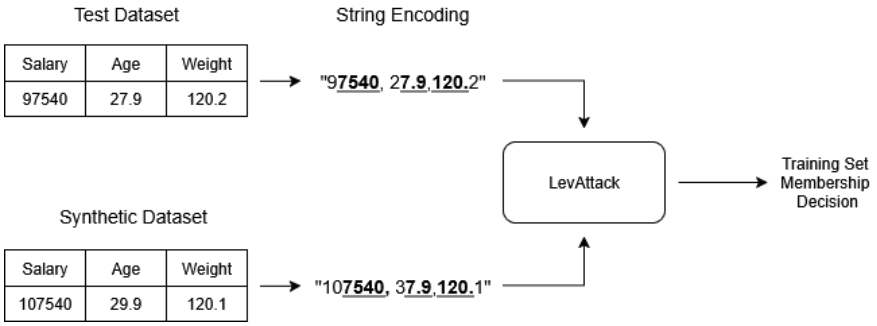


Figure 1: Diagram of Levenshtein Attack. We simply encode rows of tabular data into a string representation from which to attack. LevAtt finds signal in the highly constrained and often duplicated sequences of digits in synthetic tabular data generated by LLMs. In bold and underline: copied sequences of such patterns that are the source of LevAtt's adversarial advantage.

tain the core autoregressive formulation while differing in how they leverage available data and computational resources.

2.2 Membership Inference Attacks on Synthetic Tabular Data

Membership Inference Attacks (MIAs) aim to classify whether a specific observation was a member of the original dataset used to train a model (Shokri et al., 2017; Chen et al., 2020; Carlini et al., 2021a). Given the generative model G trained on dataset T as defined above, which generates synthetic dataset S, an adversary $\mathcal{A} : X \to \{0, 1\}$ aims to determine if a test sample x^* is an element of T. Formally, this classification or MIA can be expressed as:

$$\mathcal{A}(x^\star) = \mathbb{I}\left[f(x^\star) > \gamma\right] \tag{1}$$

where $\mathbb{I}$ is the indicator function, $f(x^\star)$ is a scoring function of the test observation x^*, and γ is an adjustable decision threshold. The success of the attack can be measured using traditional binary classification metrics and can be interpreted as a measure of privacy leakage from a model of the training data.

MIAs have become a mainstay tool to audit the privacy of tabular synthetic data as they represent a material privacy risk associated with the output of a model (Houssiau et al., 2022; Ward et al., 2025a). Indeed, a number of distance (Chen et al., 2020; Ward et al., 2024), density (Hilprecht et al., 2019; van Breugel et al., 2023), and likelihood-based (Stadler et al., 2022; Meeus et al., 2024; Ward et al., 2025b) attacks have been proposed under a wide variety of threat models. However, these existing methods all attack over the tabular feature space $\mathcal{X}^d$. LLM-based generators introduce a fundamentally different vulnerability: they generate in an intermediate string representation space before parsing to a tabular format. This autoregressive string generation process creates a novel attack surface that bypasses traditional feature-space-targeting attacks.

3 Attacking Strings of Digits

LLM-based tabular data generators operate on records encoded as sequences of characters, often representing numeric values in fixed formats. From a privacy perspective, these numeric strings constitute a **distinct threat surface**: even minor changes at the character level can correspond to meaningful alterations in sensitive information, such as financial amounts, medical measurements, or personally identifiable metrics. Unlike free-form text, where approximate matches may be semantically ambiguous, we hypothesize that numeric strings are rigid and highly informative, making them particularly vulnerable to memorization by LLMs and thus adversarial signal.

In this section, we focus on **attacks that exploit these implied strings of digits**. We show that by measuring string similarity between synthetic outputs and potential training records, an adversary can infer membership without access to the model internals, queries, or auxiliary data. This approach exposes a realistic privacy risk inherent in current LLM-based tabular data generation pipelines, emphasizing the need to examine character-level leakage beyond traditional feature-space analyses.

3.1 Threat Model

In this work, we adopt a conservative No-box threat model (Houssiau et al., 2022; Ward et al., 2025a), in which the adversary has access only to the synthetic data S produced by the model. We make no assumptions that the

adversary has knowledge of the model implementation and hyperparameters, query access, or even a reference dataset for constructing calibrated attacks. This threat model is motivated by two key considerations:

1. **Realism**. No-box threat models represent realistic privacy scenarios for tabular synthetic data practitioners. A common scenario involves a user training a tabular data generator on private datasets and sharing the generated data with public or private parties without disclosing implementation details. From the adversary's perspective, this synthetic data may be maliciously acquired or discovered on open data sharing platforms, with no additional context about the generation process.
2. **Difficulty**. No-box threat models are recognized as particularly challenging for constructing effective attacks due to the severe limitations placed on the adversary Chen et al. (2020); Houssiau et al. (2022). Specifically, the adversary cannot analyze the model's loss function, construct shadow models, or query the model directly. Successful attacks under these restrictive conditions therefore highlight fundamental vulnerabilities in these synthetic data generation methods.

3.2 Levenshtein Attack

String similarity is a well-established concept, with Levenshtein edit distance providing a fundamental measure of character-level overlap (Levenshtein, 1966; Navarro, 2001; Yujian & Bo, 2007). Recent work has applied normalized variants of this distance to quantify approximate memorization in LLMs, showing that models can reproduce training examples with minor variations in natural language text (Ippolito et al., 2023; Shilov et al., 2025). However, these studies stop short of embedding string similarity into an adversarial framework, as open-ended natural language provides many valid paraphrases and thus weak signals for membership inference.

In contrast, **the string representations of numeric values in tabular data are highly constrained:** numbers, delimiters, and fixed column formats mean that even small character-level deviations correspond to meaningful differences. This makes edit distance a far more reliable signal of memorization in our setting.

Motivated by this observation, we design an MIA based on the Distance of Closest Record (Chen et al., 2020), instantiated with Levenshtein distance. Formally, for each test observation and x^* and synthetic set S, we extract the ordered numeric and categorical values for each observation and encode them as strings (See Figure 1): x^*_{str} and S_{str}. The LevAtt score for Equation 1 is then:

$$f_{\text{LevAtt}}(x^*_{\text{str}}, S_{\text{str}}) = -\min_{s \in S_{\text{str}}} \ell(x^*_{\text{str}}, s), \tag{2}$$

where ℓ denotes the Levenshtein edit distance. Lower distances (i.e., higher scores) indicate closer matches and stronger evidence of membership. Because LevAtt requires only synthetic outputs, it applies directly to our No-box threat model highlighting a realistic privacy risk in LLM-based tabular data generation.

4 Experiments

We evaluate the privacy leakage of digit memorization for in-context learning (ICL) and supervised fine-tuning (SFT) based LLM-based tabular generators. Here, we use differing experimental designs to correspond to their popular use-case settings. For clarity, we organize the section into three subsections: experimental design details for both regimes and then separate subsections for results. We include the full experimental and implementation details in Appendix: A.

4.1 Experimental design

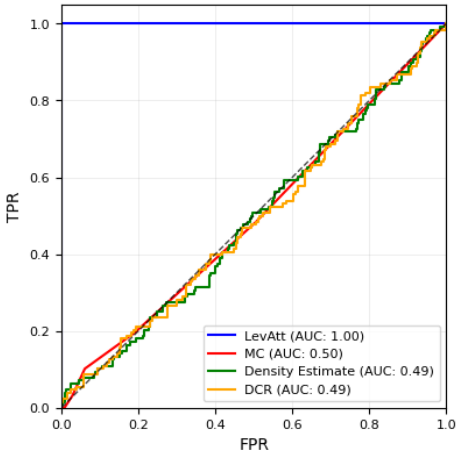
4.1.1 ICL Approaches

Replicating the design of Byun et al. (2025), we evaluate ICL approaches on the OpenML CTR23 benchmark (Fischer et al., 2023), consisting of 35 real-world classification tables with 500–100,000 rows and up to 5,000 features, including both numerical and categorical attributes. We partition each dataset into 80/20 training and test sets. To simulate the low-data regime these methods are commonly used for, we subsample $32, 64, 128$ training rows without replacement. These sampled rows are provided as exemplars to the ICL models.

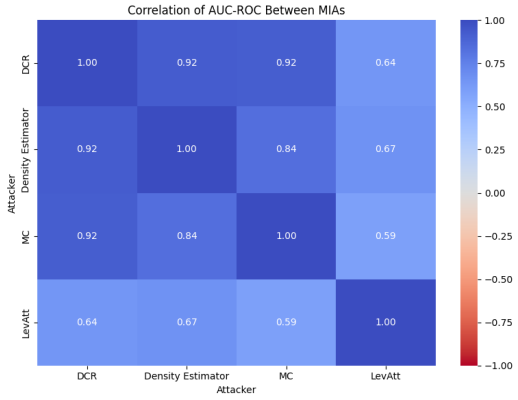
We consider three ICL models: LLaMA 3.3 70B (Meta AI, 2024) an open-source foundation model, GPT-4o-mini (OpenAI, 2024) a close-source foundation model, and TabPFN-v2 Hollmann et al. (2025a) a transformer-based model trained on tabular data due to their wide use and availability. For each sampled training subset, we generate an equal amount of synthetic data. Evaluation is conducted on all training rows plus an equal holdout partition of the test set. We evaluate privacy leakage using the following No-box MIAs using the

Table 1: LevAtt performance against ICL models. Mean (STD) values are reported for all datasets, training sizes, and seeds (Overall) and for the top 20 runs for each model with the highest AUC-ROC (Top 20). GPT-4o-mini shows relatively little privacy leakage, whereas LLama-3.3-70b and TabPFN-V2 show significant privacy failure- especially amongst their worst datasets.

Generator	AUC-ROC		TPR@FPR=0		TPR@FPR=0.1	
	Overall	Top 20	Overall	Top 20	Overall	Top 20
LLama-3.3-70b	**0.63 (0.12)**	**0.91 (0.03)**	**0.08 (0.20)**	**0.64 (0.27)**	**0.28 (0.21)**	0.81 (0.09)
TabPFN-V2	0.58 (0.05)	0.91 (0.07)	0.04 (0.01)	0.57 (0.41)	0.19 (0.06)	**0.94 (0.14)**
GPT-4o-mini	0.54 (0.05)	0.60 (0.09)	0.01 (0.01)	0.01 (0.03)	0.13 (0.05)	0.27 (0.09)



(a) ROC plot for various No-box MIAs against TabPFN-V2 with 128 in-context samples from the MoneyBall dataset. LevAtt (blue) is able to achieve perfect classification for all in-context samples whereas MIAs that target the feature space of tabular data fail to capture the privacy leakage.



(b) Correlation plot for No-box MIA AUC-ROC across the ICL experiment. While the feature-space targeting DCR, Density Estimate, and MC are nearly perfectly correlated, LevAtt is much less correlated. This highlights that while privacy leakage over tabular string representations and the feature space are related, LevAtt finds unique adversarial advantage.

Synth-MIA package (Ward et al., 2025a): LevAtt, Euclidean Distance to Closest Record (DCR) (Chen et al., 2020), a Monte Carlo density estimation (MC) method (Hilprecht et al., 2019), and a kernel density estimation method (Houssiau et al., 2022; van Breugel et al., 2023). We report MIA success using mean AUC-ROC and TPR at fixed FPR thresholds (Carlini et al., 2021a) to capture potential information leakage. We repeat this experimentation across 3 seeded runs.

4.1.2 SFT Approaches

For SFT-approaches, we benchmark the original SFT-based tabular generation method GREAT Borisov et al. (2023) and RealTabFormer (Solatorio & Dupriez, 2023), which reports improved privacy performance due to enforcing a minimum Euclidean Distance to Closest to Record distribution in its training. As both of these methods use GPT-2 (Radford et al., 2019)as a base model, we modify RealTabFormer to accept more modern, larger foundation models: LLaMA 3.2 (1B, 3B) (Grattafiori et al., 2024), Qwen2.5-3B (Qwen et al., 2025), and Mistral v0.3 7B Jiang et al. (2023). Additionally, we introduce as a control CT-GAN and TVAE Xu et al. (2019), conventional deep learning-based generators. Training follows default hyperparameters from the original GREAT and RealTabFormer implementations while CTGAN and TVAE are implemented through Synthcity Qian et al. (2023).

Experiments are conducted on five tabular datasets: CASP, Abalone, Diabetes, CA-Housing, and Faults, selected for their common use in synthetic tabular data benchmarking and containing numeric data. Similarly to ICL, we create 80/20 train-test partitions and following common synthetic tabular data benchmarking (Zhang et al., 2024), synthetic datasets equal in size to the original training sets are generated post-training. For privacy evaluation, up to 1,000 training and holdout samples are partitioned as test sets, and the same MIAs are applied. Experiments are repeated across three seeds.

Table 2: Mean (STD) LevAtt performance for each model and dataset across seeds. RealTabFormer experiences significant privacy leakage and LevAtt finds some signal for most LLM-based models. However, LevAtt identifies no leakage in conventional deep learning-based methods CT-GAN and TVAE as they do not generate strings.

Model	LevAtt Metric	CA-Housing	CASP	Abalone	Diabetes	Faults
RealTabFormer	AUC-ROC	**0.70 (0.11)**	**0.72 (0.12)**	0.60 (0.08)	**0.66 (0.04)**	**0.61 (0.12)**
	TPR@FPR=0.1	**0.34 (0.23)**	**0.37 (0.11)**	0.21 (0.05)	0.25 (0.05)	**0.32 (0.10)**
LLaMA 3.2-1B	AUC-ROC	0.68 (0.22)	0.52 (0.15)	0.50 (0.03)	0.56 (0.03)	0.58 (0.15)
	TPR@FPR=0.1	0.21 (0.21)	0.10 (0.06)	0.10 (0.02)	0.21 (0.02)	0.16 (0.05)
LLaMA 3.2-3B	AUC-ROC	0.63 (0.06)	0.50 (0.07)	**0.62 (0.04)**	0.61 (0.04)	0.58 (0.07)
	TPR@FPR=0.1	0.24 (0.09)	0.09 (0.03)	**0.22 (0.11)**	0.15 (0.11)	0.16 (0.03)
Qwen 2.5-3B	AUC-ROC	0.61 (0.08)	0.54 (0.02)	0.52 (0.04)	0.64 (0.04)	0.56 (0.01)
	TPR@FPR=0.1	0.25 (0.05)	0.12 (0.02)	0.15 (0.07)	**0.28 (0.07)**	0.11 (0.02)
Mistral-7B	AUC-ROC	0.67 (0.13)	0.51 (0.01)	0.51 (0.03)	0.53 (0.06)	0.52 (0.01)
	TPR@FPR=0.1	0.23 (0.04)	0.11 (0.02)	0.10 (0.05)	0.10 (0.03)	0.10 (0.01)
GREAT	AUC-ROC	0.66 (0.09)	0.55 (0.04)	0.52 (0.03)	0.50 (0.05)	0.54 (0.07)
	TPR@FPR=0.1	0.33 (0.10)	0.14 (0.03)	0.11 (0.02)	0.10 (0.03)	0.14 (0.02)
CT-GAN	AUC-ROC	0.48 (0.03)	0.51 (0.04)	0.48 (0.03)	0.47 (0.04)	0.50 (0.02)
	TPR@FPR=0.1	0.09 (0.01)	0.11 (0.01)	0.12 (0.01)	0.10 (0.01)	0.08 (0.01)
TVAE	AUC-ROC	0.47 (0.03)	0.49 (0.02)	0.51 (0.03)	0.53 (0.02)	0.51 (0.03)
	TPR@FPR=0.1	0.08 (0.01)	0.10 (0.01)	0.11 (0.01)	0.10 (0.01)	0.10 (0.02)

4.2 Results: ICL Methods

We summarize our key findings for our ICL experiments as follows:

LevAtt Identifies Substantial Privacy Leakage in ICL Tabular Synthetic Data Approaches. We report the Mean (STD) for LevAtt's AUC-ROC and TPR@FPR = $\{0.0, 0.1\}$ over all datasets, training sizes, and seeds in Table 1. As there is considerable variability in the performance of LevAtt across runs and MIAs are often concerned with the worst-case performance of a model, we also report corresponding values for the top 20 runs for each model by AUC-ROC. Here, LLama 3.3-70b and TabPFN-V2 experience substantial privacy leakage based on memorizing approximate patterns in the training data with mean TPR@FPR=0 values of 0.64 and 0.57 respectively. In some cases, we even find that LevAtt is a perfect classifier of training membership (see Figure 1a).

Privacy Leakage Likely Scales with Model Size. Prior work (Carlini et al., 2021b; 2023) has shown that as the parameter count of LLMs increases, so does their capacity for memorization. While the exact sizes of GPT-4o-mini and TabPFN-V2 are not publicly disclosed, benchmarking suggests GPT-4o-mini performs similarly to models with roughly 7B parameters, and TabPFN-V2 is small enough to run on a single GPU. In contrast, the 70B-parameter LLama 3.3 exhibits markedly higher privacy leakage on our benchmark, consistent with the expected scaling behavior.

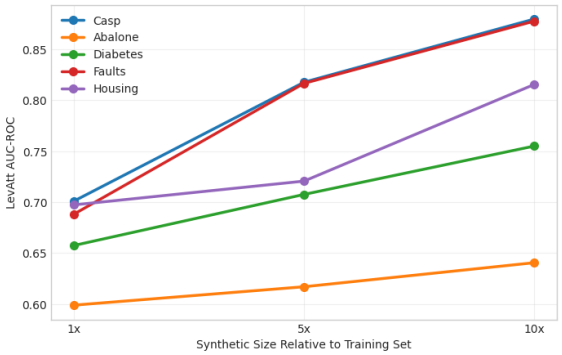
LevAtt Captures a Unique String-Similarity Signal Relative to Other MIAs. Figure 1b shows the correlation of AUC-ROC for LevAtt with other No-box threat model MIAs targeting the feature space of synthetic data—DCR, Density Estimate, and MC—across all datasets, training sizes, seeds, and models. Feature-space attacks are nearly perfectly correlated with each other, as expected. In contrast, LevAtt exhibits lower correlation, suggesting that it uncovers privacy leakage signals not captured by traditional feature-space attacks.

4.3 Results: SFT Methods

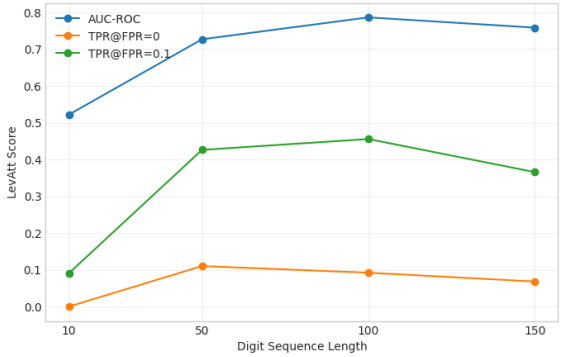
We summarize our key findings for our SFT experiments as follows:

While Less Sizeable, SFT-Approaches Similarly Exhibit Privacy Leakage. We report the Mean (STD) of Lev-Att's AUC-ROC and TPR@FPR=0.1 across datasets and seeds for each model Table in 2. Overall, SFT-based approaches generally exhibit lower privacy leakage than ICL counterparts. Surprisingly, RealTabFormer often shows the highest LevAtt scores, despite using privacy-aware training and sharing a GPT-2 base with GREAT, which typically leaks little. LevAtt also detects leakage in LLaMA-3.2 (1B and 3B) and Qwen-2.5-3B. In contrast, LevAtt is completely ineffective against CT-GAN and TVAE, as these models generate tabular data at the level of complete observations rather than token by token, inherently protecting against this type of attack.

LevAtt Success is Related to the Synthetic Dataset Size. To examine how synthetic data volume affects privacy leakage, we conducted an ablation study where RealTabFormer generated synthetic datasets at 1×, 5×, and 10× the size of each original training dataset. The results in Figure 2a reveal a clear monotonic relationship: LevAtt AUC-ROC scores increase consistently with larger synthetic datasets, with the Faults dataset showing



(a) LevAtt AUC-ROC for various datasets generated by RealTabFormer with increasing synthetic dataset sizes relative to the training set.



(b) LevAtt performance on RealTabFormer at various training digit sequence lengths.

a particularly striking 20% improvement in attack success. This pattern demonstrates that generating more synthetic data systematically increases the probability of reproducing memorized training examples, thereby amplifying privacy risks.

LevAtt Success is Related to the Sequence Length of Training Data. As an additional ablation, we investigate the impact of digit sequence length in the training data. Here, we generate both training and holdout datasets from the same multivariate standard Gaussian distribution each with 10,000 observations. To control for sequence length, we select an increasing number of columns for the training data, each containing exactly 10 digits. We then train RealTabFormer on datasets with progressively longer digit sequences to assess its effect on model behavior. We report LevAtt's performance in Figure 2b where we see that increasing the count of digits or sequence length in the training data correlates with increased memorization in RealTabFormer. This further replicates findings from Carlini et al. (2021b) that found training sequence length factors into increased memorization in LLMs.

5 Defenses Against Levenshtein Attack

The efficacy of LevAtt against LLM-based tabular models motivates us to explore methodologies that can defend generated synthetic data. Here, recognizing that LevAtt gains adversarial advantage from replicated patterns in strings of digits we devise two strategies based on introducing controlled "noise" into string sequences: Digit Modifier (DM) a post-processing method that operates independently of the generative model and is applied after the synthetic data have been produced, and Tendency-based Logit Processor (TLP) a method compatible with any open-source LLM that strategically perturbs digits at sample time.

5.1 Digit Modifier and Tendency-Based Logit Processor

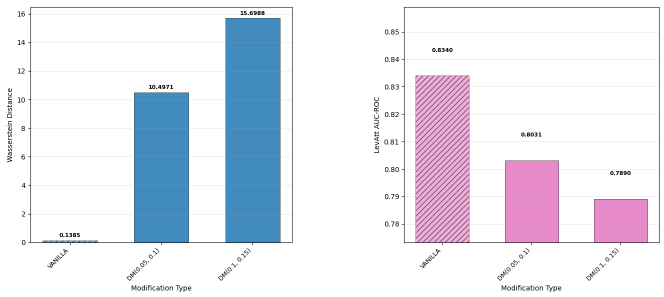
Digit Modifier

We propose a Digit Modifier (DM), a method for injecting controlled randomness into tabular data by perturbing numerical digits. Motivated by bit-flipping techniques for adding noise to binary representations of relational databases Agrawal & Kiernan (2002); Alfagi et al. (2016), DM operates by replacing tokens in a record's tokenized sequence with alternatives sampled from a noise distribution. When tokens represent numerical values, these substitutions correspondingly alter the digits of the underlying sequences, yielding perturbed records. In this sense, DM can be viewed as a principled method for randomly replacing digits within a record.

To balance data fidelity and protection, we parameterize the mechanism as $\text{DM}(p_{min}, p_{max})$ with $0 \leq p_{min} < p_{max} \leq 1$. For a numerical column $\mathbf{X_i}$ and entry $\mathbf{x_{k,i}}$, a probability function $g : \mathbf{x_{k,i}} \times \mathbf{X_i} \to [p_{min}, p_{max}]$ assigns digit-flipping probabilities such that larger-magnitude entries receive higher perturbation probabilities. Each digit of $\mathbf{x_{k,i}}$ is then independently flipped according to $g(\mathbf{x_{k,i}}, \mathbf{X_i})$, while smaller-magnitude values—being more sensitive—undergo smaller changes to preserve fidelity. The design of the probability function g and full algorithm (Alg 1) are described in Appendix B.1.

Tendency-Based Logit Processor

We additionally propose Tendency-based Logit Processor (TLP), a mechanism for injecting controlled noise into synthetic data by perturbing the generator's logits at inference time. The TLP(t) selectively amplifies lower-



(a) Wasserstein distances between training and synthetic datasets.

(b) LevAtt AUC-ROC of synthetic datasets evaluated with an equal number of training and non-training records.

Figure 3: Privacy–fidelity comparison of DM on RealTabFormer synthetic data from the simulated Gaussian dataset (Section 5.2). VANILLA corresponds to plain sampling without protection. Panel (a) reports Wasserstein distances; panel (b) shows LevAtt AUC-ROC. While DM is able to induce reductions in privacy leakage, the resulting synthetic data are of low fidelity.

valued logits while suppressing higher-valued ones, making the generator more likely to select tokens that were originally less probable. The strength of this effect is controlled by the tendency parameter t: higher values of t induce stronger curvature, increasing the randomness of the generated sequence. In this way, TLP(t) acts as a principled method for introducing variability into synthetic outputs while still preserving the generator's learned distribution.

Formally, TLP(t) first maps the generator's logits $l = (l_1, l_2, \ldots, l_k)$ into the range $[0, 1]$ using a shifted min–max scaler S_l. It then applies a monotone increasing, concave curving function $f_t : [0, 1] \to [0, 1]$, parameterized by t and satisfying $f_t(0) = 0$, to each scaled logit. Finally, the processed logits are transformed back to their original scale using the inverse scaler S_l^{-1}.

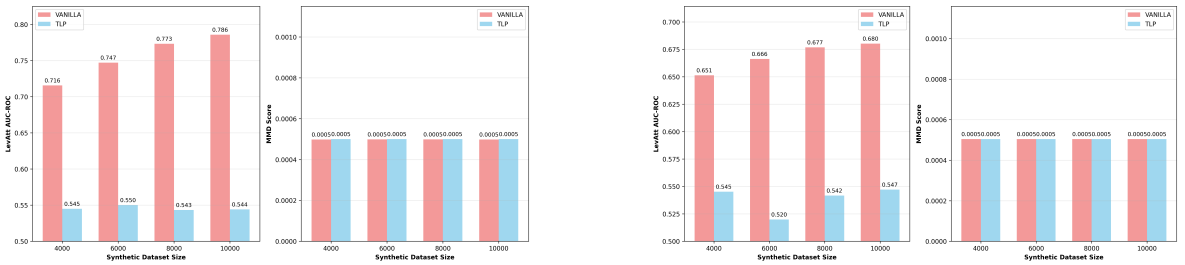
The design of f_t is central to TLP(t). Monotonicity ensures that the relative order of logits is preserved, so high-probability tokens remain more likely than lower-probability ones, allowing controlled noise injection without overwhelming the generator's learned signal. Concavity, combined with $f_t(0) = 0$, guarantees that lower logits are curved upward, increasing their chance of being sampled. Appendix B.2 provides full details of the scalers S_l and S_l^{-1}, the conditions for valid curving, the specific f_t used in our experiments, and a full algorithm description (Alg 2).

5.2 Evaluating Digit Modifier and Tendency-Based Logit Processor

To evaluate the effectiveness of DM and TLP, we use RealTabFormer, the SFT model exhibiting the highest degree of memorization, and measure how much privacy improvement each method provides against LevAtt while preserving synthetic data fidelity. Following the experimental designs of Section 4.3, we generate a 20-column training and holdout set from a Multivariate Gaussian N(300, 5), and we use the CASP dataset at varying synthetic set sizes to induce privacy leakage. We then apply DM and TLP across different scaler and tendency parameter levels respectively, and assess performance in terms of privacy (LevAtt AUC-ROC), fidelity (Wasserstein Distance and Maximum Mean Discrepancy between training and synthetic datasets), and utility (RMSE of XGBoost models trained on synthetic data and evaluated on real holdout data) (Qian et al., 2023).

Overall, we find that while effective in reducing privacy leakage, DM suffers large fidelity costs (See Figure 3). DM is particularly challenged by low-bandwidth datasets or columns with restricted ranges, where naive modifications can shift values into low-density regions and result in significant fidelity loss. On the other hand, TLP, with a tuned tendency parameter t, can controllably reduces attack efficacy while effectively preserving the fidelity between the processed synthetic data and the training data. In Figure 4, in both highly unprivate Simulation and CASP datasets TLP reduces LevAtt's AUC-ROC to 55% with effectively no penalty in the Maximum Mean Discrepancy of TLP-generated data.

We include additional experiments for TLP in Appendix B.3. There, we show that TLP is able to also control for TPR@FPR values with minimal fidelity loss (see Figure 6) as well as preserve utility in machine learning models (See Figure 7). Overall, TLP offers a competive defense against string-based attacks.



(a) Simulation Dataset

(b) CASP Dataset

Figure 4: Privacy–fidelity trade-off of TLP in RealTabFormer synthetic data from the simulation (Section 5.2) and CASP dataset. For both simulation and CASP sets, TLP reduces LevAtt AUC–ROC to 55% across synthetic dataset sizes, improving privacy over the vanilla inference, while the Maximum Mean Discrepancy (MMD) remains nearly unchanged, demonstrating that TLP preserves fidelity while mitigating privacy leakage.

6 Discussion

LevAtt reveals that LLM-based tabular data generation is uniquely unsafe relative to conventional deep learning approaches. While being a simple string-similarity attack operating under an extremely restrictive threat model, LevAtt uncovered that state-of-the-art models could catastrophically leak training membership by copying sequential patterns of digits and text from training examples. This phenomenon was further demonstrated in SFT-generators, revealing that the base implementation of RealTabFormer—a method recognized for its privacy-preserving capabilities—was susceptible to significant privacy leakage. At the same time, LevAtt was found to be substantially less correlated with feature space–oriented MIAs, and conventional tabular data generators such as CT-GAN and TVAE proved resistant to our string-based attack. These findings suggest that tabular models relying on autoregressive token-based generation expose a distinct attack surface that prior deep-learning architectures do not share.

While highly deployable and powerful, LevAtt can be defeated. In this work, we proposed an intuitive post-hoc defense called Digit Modifier (DM), which alters digits after generation. However, we found that DM failed to preserve the fidelity of the synthetic dataset. This limitation motivated the development of TLP, which can be appended to any open-source LLM through the Hugging Face API. TLP can effectively control for privacy leakage from LevAtt by smoothing the logits of digits with disproportionately high probabilities. We found that this perturbation did not substantially alter the statistical fidelity of the resulting synthetic data.

While TLP provides an effective defense against LevAtt, the need for such inference-time mitigation raises a broader question: *What are these models exactly learning?* A prevailing assumption is that LLM-based tabular generators approximate the joint distribution of the training set T through sequential string modeling, similar to conventional generative approaches. However, our findings suggest that in some cases these models behave more like perturbation mechanisms, producing outputs that closely resemble training examples with only minor modifications. This behavior naturally yields high fidelity and downstream utility but does so precisely because of its proximity to the training data—thereby exposing the system to privacy leakage. Understanding when LLMs genuinely learn tabular distributions versus when they rely on approximate memorization remains an important direction for future work.

7 Conclusion

In this paper we introduce LevAtt, a No-box threat model MIA that exposes substantial privacy risk for in-context learning and supervised-finetuned LLM-based tabular data generators. LevAtt shows that LLMs are vulnerable to memorization from the structured, often duplicated patterns of tabular data. By attacking the string encodings of autoregressively generated tabular data, LevAtt finds unique adversarial signal compared to existing methods. While less restrictive threat models would likely lead to a better attack, we believe No-box carries a powerful message: an attack with the minimal assumptions reasonably possible for an adversary can perfectly classify training membership in state-of-the-art generators. Lastly propose two defenses against LevAtt, showing that Tendency-Based Logit Processor can effectively defeat LevAtt with minimal loss in synthetic data fidelity. Future research directions could involve developing even more powerful attacks under less restrictive threat models, finding more efficient and provable defenses, and studying mechanistically how LLMs represent tabular distributions.

8 Statement of Ethics

The potential for adversaries to determine whether an individual's data was included in the original dataset presents significant privacy risks, especially in fields such as healthcare, finance, and social sciences, where sensitive personal information is commonly used. Synthetic data that fails to sufficiently mask membership information could inadvertently enable re-identification. Although this work introduces a method for assessing such risks, its primary objective is to empower researchers and practitioners to perform more rigorous privacy evaluations before deploying synthetic datasets. We emphasize that adversarial approaches are essential for advancing the development of robust privacy-preserving systems.

9 Reproducibility Statement

Our code is available through the link provided on the abstract page. The main paper explains all algorithms in detail and provides the dataset and simulation descriptions, while model hyperparameters are reported in the appendix.

References

Rakesh Agrawal and Jerry Kiernan. Watermarking relational databases. In *VLDB'02: Proceedings of the 28th International Conference on Very Large Databases*, pp. 155–166. Elsevier, 2002.

Abd S Alfagi, A Abd Manaf, B Hamida, S Khan, and Ali A Elrowayati. Survey on relational database watermarking techniques. *ARPN-JEAS*, 11:422–423, 2016.

Ankur Ankan and Abinash Panda. pgmpy: Probabilistic graphical models using python. In *Proceedings of the Python in Science Conference*, SciPy. SciPy, 2015. doi: 10.25080/majora-7b98e3ed-001. URL http://dx.doi.org/10.25080/Majora-7b98e3ed-001.

Samuel A. Assefa, Danial Dervovic, Mahmoud Mahfouz, Robert E. Tillman, Prashant Reddy, and Manuela Veloso. Generating synthetic data in finance: opportunities, challenges and pitfalls. In *Proceedings of the First ACM International Conference on AI in Finance*, ICAIF '20, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450375849. doi: 10.1145/3383455.3422554. URL https://doi.org/10.1145/3383455.3422554.

Vadim Borisov, Kathrin Seßler, Tobias Leemann, Martin Pawelczyk, and Gjergji Kasneci. Language models are realistic tabular data generators, 2023. URL https://arxiv.org/abs/2210.06280.

Jessup Byun, Xiaofeng Lin, Joshua Ward, and Guang Cheng. Risk in context: Benchmarking privacy leakage of foundation models in synthetic tabular data generation, 2025. URL https://arxiv.org/abs/2507.17066.

Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, A. Terzis, and Florian Tramèr. Membership inference attacks from first principles, 2021a. URL https://api.semanticscholar.org/CorpusID:244920593.

Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, Alina Oprea, and Colin Raffel. Extracting training data from large language models, 2021b. URL https://arxiv.org/abs/2012.07805.

Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramer, and Chiyuan Zhang. Quantifying memorization across neural language models, 2023. URL https://arxiv.org/abs/2202.07646.

Dingfan Chen, Ning Yu, Yang Zhang, and Mario Fritz. Gan-leaks: A taxonomy of membership inference attacks against generative models. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, CCS '20, pp. 343–362, Virtual Event, USA, October 2020. ACM. doi: 10.1145/3372297.3417238. URL http://dx.doi.org/10.1145/3372297.3417238.

Tianzhe Chu, Yuexiang Zhai, Jihan Yang, Shengbang Tong, Saining Xie, Dale Schuurmans, Quoc V Le, Sergey Levine, and Yi Ma. SFT memorizes, RL generalizes: A comparative study of foundation model post-training. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=dYur3yabMj.

Conor Durkan, Artur Bekasov, Iain Murray, and George Papamakarios. Neural spline flows. In *Advances in Neural Information Processing Systems*, volume 32, pp. 7627–7638, Vancouver, Canada, 2019. Curran Associates Inc.

Sebastian Felix Fischer, Matthias Feurer, and Bernd Bischl. OpenML-CTR23 – a curated tabular regression benchmarking suite. In *AutoML Conference 2023 (Workshop)*, 2023. URL https://openreview.net/forum?id=HebAOoMm94.

Brendan Flanagan, Rwitajit Majumdar, and Hiroaki Ogata. Fine grain synthetic educational data: Challenges and limitations of collaborative learning analytics. *IEEE Access*, 10:26230–26241, 03 2022. doi: 10.1109/ACCESS.2022.3156073.

Joao Fonseca and Fernando Bação. Tabular and latent space synthetic data generation: a literature review. *Journal of Big Data*, 10, 07 2023. doi: 10.1186/s40537-023-00792-7.

Mauro Giuffré and Dennis L. Shung. Harnessing the power of synthetic data in healthcare: innovation, application, and privacy. *NPJ Digital Medicine*, 6, 2023. URL https://api.semanticscholar.org/CorpusID:263802405.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vítor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu

Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Benjamin Hilprecht, Martin Härterich, and Daniel Bernau. Monte carlo and reconstruction membership inference attacks against generative models. *Proceedings on Privacy Enhancing Technologies*, 2019:232 – 249, 2019. URL https://api.semanticscholar.org/CorpusID:199546273.

Noah Hollmann, Samuel Müller, Lennart Purucker, Arjun Krishnakumar, Max Körfer, Shi Bin Hoo, Robin Tibor Schirrmeister, and Frank Hutter. Accurate predictions on small data with a tabular foundation model. *Nature*, 637(8045):319–326, 2025a.

Noah Hollmann, Samuel Müller, Lennart Purucker, Arjun Krishnakumar, Max Körfer, Shi Hoo, Robin Schirrmeister, and Frank Hutter. Accurate predictions on small data with a tabular foundation model. *Nature*, 637:319–326, 01 2025b. doi: 10.1038/s41586-024-08328-6.

Florimond Houssiau, James Jordon, Samuel N Cohen, Owen Daniel, Andrew Elliott, James Geddes, Callum Mole, Camila Rangel-Smith, and Lukasz Szpruch. Tapas: a toolbox for adversarial privacy auditing of synthetic data, 2022.

Daphne Ippolito, Florian Tramer, Milad Nasr, Chiyuan Zhang, Matthew Jagielski, Katherine Lee, Christopher Choquette Choo, and Nicholas Carlini. Preventing generation of verbatim memorization in language models gives a false sense of privacy. In C. Maria Keet, Hung-Yi Lee, and Sina Zarrieß (eds.), *Proceedings of the 16th International Natural Language Generation Conference*, pp. 28–53, Prague, Czechia, September 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.inlg-main.3. URL https://aclanthology.org/2023.inlg-main.3/.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b, 2023. URL https://arxiv.org/abs/2310.06825.

Nikhil Kandpal, Eric Wallace, and Colin Raffel. Deduplicating training data mitigates privacy risks in language models. *ArXiv*, abs/2202.06539, 2022. URL https://api.semanticscholar.org/CorpusID:246823128.

Jinhee Kim, Taesung Kim, and Jaegul Choo. EPIC: Effective prompting for imbalanced-class data synthesis in tabular data classification via large language models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=d5cKDHCrFJ.

Akim Kotelnikov, Dmitry Baranchuk, Ivan Rubachev, and Artem Babenko. Tabddpm: Modelling tabular data with diffusion models, 2022.

Vladimir I Levenshtein. Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physics Doklady*, 10:707, February 1966.

Junwei Ma, Apoorv Dankar, George Stein, Guangwei Yu, and Anthony Caterini. Tabpfgen – tabular data generation with tabpfn, 2024. URL https://arxiv.org/abs/2406.05216.

Ryan McKenna, Brett Mullins, Daniel Sheldon, and Gerome Miklau. Aim: an adaptive and iterative mechanism for differentially private synthetic data. *Proc. VLDB Endow.*, 15(11):2599–2612, July 2022. ISSN 2150-8097. doi: 10.14778/3551793.3551817. URL https://doi.org/10.14778/3551793.3551817.

Matthieu Meeus, Florent Guepin, Ana-Maria Creţu, and Yves-Alexandre de Montjoye. *Achilles' Heels: Vulnerable Record Identification in Synthetic Data Publishing*, pp. 380–399. Springer Nature Switzerland, Cham, Switzerland, 2024. ISBN 9783031514760. doi: 10.1007/978-3-031-51476-0_19. URL http://dx.doi.org/10.1007/978-3-031-51476-0_19.

Meta AI. LLaMA-3.3 70B instruct model. https://huggingface.co/meta-llama/Llama-3.3-70B-Instruct, 2024. Released December 6, 2024; accessed 2025-06-13.

Gonzalo Navarro. A guided tour to approximate string matching. *ACM Computing Surveys*, 33(1):31–88, 2001.

OpenAI. GPT-4o Mini model in chat completions api. https://platform.openai.com/docs/models/gpt-4o-mini, 2024. Released July 18, 2024; accessed 2025-06-13.

Michael Platzer and Thomas Reutterer. Holdout-based empirical assessment of mixed-type synthetic data. *Frontiers in big Data*, 4:679939, 2021.

Zhaozhi Qian, Bogdan-Constantin Cebere, and Mihaela van der Schaar. Synthcity: facilitating innovative use cases of synthetic data in different data modalities, 2023. URL https://arxiv.org/abs/2301.07573.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025. URL https://arxiv.org/abs/2412.15115.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

Nabeel Seedat, Nicolas Huynh, Boris van Breugel, and Mihaela van der Schaar. Curated LLM: Synergy of LLMs and data curation for tabular augmentation in low-data regimes. In *Forty-first International Conference on Machine Learning*, 2024.

Igor Shilov, Matthieu Meeus, and Yves-Alexandre de Montjoye. The mosaic memory of large language models, 2025. URL https://arxiv.org/abs/2405.15523.

R. Shokri, M. Stronati, C. Song, and V. Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy (SP)*, pp. 3–18, Los Alamitos, CA, USA, may 2017. IEEE Computer Society. doi: 10.1109/SP.2017.41. URL https://doi.ieeecomputersociety.org/10.1109/SP.2017.41.

Aivin V Solatorio and Olivier Dupriez. Realtabformer: Generating realistic relational and tabular data using transformers. *arXiv preprint arXiv:2302.02041*, 2023.

Theresa Stadler, Bristena Oprisanu, and Carmela Troncoso. Synthetic data – anonymisation groundhog day. In *31st USENIX Security Symposium (USENIX Security 22)*, pp. 1451–1468, Boston, MA, August 2022. USENIX Association. ISBN 978-1-939133-31-1. URL https://www.usenix.org/conference/usenixsecurity22/presentation/stadler.

Namjoon Suh, Xiaofeng Lin, Din-Yin Hsieh, Mehrdad Honarkhah, and Guang Cheng. Autodiff: combining auto-encoder and diffusion model for tabular data synthesizing, 2023. URL https://openreview.net/forum?id=XhxOCXlXSh.

Vibeke Binz Vallevik, Aleksandar Babic, Serena E. Marshall, Severin Elvatun, Helga M.B. Brøgger, Sharmini Alagaratnam, Bjørn Edwin, Narasimha R. Veeraragavan, Anne Kjersti Befring, and Jan F. Nygård. Can i trust my fake data – a comprehensive quality assessment framework for synthetic tabular data in healthcare. *International Journal of Medical Informatics*, 185:105413, 2024. ISSN 1386-5056. doi: https://doi.org/10.1016/j.ijmedinf.2024.105413. URL https://www.sciencedirect.com/science/article/pii/S1386505624000765.

Boris van Breugel, Hao Sun, Zhaozhi Qian, and Mihaela van der Schaar. Membership inference attacks against synthetic data through overfitting detection, 2023.

Yuxin Wang, Duanyu Feng, Yongfu Dai, Zhengyu Chen, Jimin Huang, Sophia Ananiadou, Qianqian Xie, and Hao Wang. Harmonic: harnessing llms for tabular data synthesis and privacy protection. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, NIPS '24, Red Hook, NY, USA, 2025. Curran Associates Inc. ISBN 9798331314385.

Zhepeng Wang, Runxue Bao, Yawen Wu, Jackson Taylor, Cao Xiao, Feng Zheng, Weiwen Jiang, Shangqian Gao, and Yanfu Zhang. Unlocking memorization in large language models with dynamic soft prompting. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 9782–9796, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.546. URL https://aclanthology.org/2024.emnlp-main.546/.

Joshua Ward, Chi-Hua Wang, and Guang Cheng. Data plagiarism index: Characterizing the privacy risk of data-copying in tabular generative models, 2024. URL https://arxiv.org/abs/2406.13012.

Joshua Ward, Xiaofeng Lin, , Chi-Hua Wang, and Guang Cheng. Synth-mia: A testbed for auditing privacy leakage in tabular data synthesis, 2025a. URL https://arxiv.org/abs/2509.18014.

Joshua Ward, Chi-Hua Wang, and Guang Cheng. Privacy auditing synthetic data release through local likelihood attacks, 2025b. URL https://arxiv.org/abs/2508.21146.

David S. Watson, Kristin Blesch, Jan Kapar, and Marvin N. Wright. Adversarial random forests for density estimation and generative modeling. In Francisco Ruiz, Jennifer Dy, and Jan-Willem van de Meent (eds.), *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of Machine Learning Research*, pp. 5357–5375, Valencia, Spain, 25–27 Apr 2023. PMLR. URL https://proceedings.mlr.press/v206/watson23a.html.

Jinhong Wu, Konstantinos Plataniotis, Lucy Liu, Ehsan Amjadian, and Yuri Lawryshyn. Interpretation for variational autoencoder used to generate financial synthetic tabular data. *Algorithms*, 16:121, 02 2023. doi: 10.3390/a16020121.

Lei Xu, Maria Skoularidou, Alfredo Cuesta-Infante, and Kalyan Veeramachaneni. Modeling tabular data using conditional gan. In *Advances in Neural Information Processing Systems*, volume 32, pp. 7335–7345, Vancouver, Canada, 2019. Curran Associates, Inc. URL https://proceedings.neurips.cc/paper/2019/hash/254ed7d2de3b23ab10936522dd547b78-Abstract.html.

Jinsung Yoon, James Jordon, and Mihaela van der Schaar. PATE-GAN: Generating synthetic data with differential privacy guarantees. In *International Conference on Learning Representations*, pp. 1–15, New Orleans, LA, USA, May 2019. OpenReview.net. URL https://openreview.net/forum?id=S1zk9iRqF7.

Jinsung Yoon, Lydia N Drumright, and Mihaela Van Der Schaar. Anonymization through data synthesis using generative adversarial networks (ads-gan). *IEEE journal of biomedical and health informatics*, 24(8):2378–2388, 2020.

Li Yujian and Liu Bo. A normalized levenshtein distance metric. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(6):1091–1095, June 2007. ISSN 0162-8828. doi: 10.1109/TPAMI.2007.1078. URL https://doi.org/10.1109/TPAMI.2007.1078.

Hengrui Zhang, Jiani Zhang, Zhengyuan Shen, Balasubramaniam Srinivasan, Xiao Qin, Christos Faloutsos, Huzefa Rangwala, and George Karypis. Mixed-type tabular data synthesis with score-based diffusion in latent space. In *The Twelfth International Conference on Learning Representations*, pp. 4Ay23yeuz0, Vienna, Austria, May 2024. OpenReview.net. URL https://openreview.net/forum?id=4Ay23yeuz0.

A Section 4 Experiment Details

A.1 In-Context Learning Details

A.1.1 Generator Details

TabPFN Generation Hollmann et al. (2025a): We followed the Prior Labs tutorial (https://priorlabs.ai/tutorials/unsupervised/) for unsupervised TabPFN. Each training split was loaded, shuffled, and divided into batches of 200 rows. Numeric features were cast to `float32`, while categorical variables were label-encoded (assigning unseen categories to –1). Columns with zero variance were removed prior to model fitting and reintroduced after sampling. For each batch, we fit TabPFN and generated synthetic rows using temperature $t = 1.0$ across three random permutations. Outputs were decoded, constant columns reattached, batches concatenated, and finally truncated to match the size of the original dataset.

LLaMA Generation Meta AI (2024): We used LLaMA 3.3 70B via the Groq API (https://console.groq.com/docs/models). Each training split was divided into batches of up to 32 rows, ensuring that all rows were fully included in the prompt. For each batch, we computed per-column summary statistics and serialized the data to CSV. We then queried `llama-3.3-70b-versatile` with temperature $t = 1.0$, requesting N rows in JSON format. If outputs contained parse errors or incorrect row counts, we retried the generation up to five times. Valid generations were concatenated, truncated, or re-prompted as necessary, and validated for type and dimensional consistency.[2]

GPT-4o-mini OpenAI (2024): We applied the same prompting and inference pipeline as with LLaMA 3.3 70B. However, we used OpenAI's structured output API, defining the target format as a JSON schema with column names as keys and corresponding cell values as entries.

A.1.2 Prompt Template

Listing 1: Prompt passed to Groq API

```
System role: You are a tabular synthetic data generation model.

Your goal is to produce data that mirrors the given examples in
causal structure and feature/label distributions,
while maximizing diversity.

Context: Leverage your in-context learning to generate realistic,
diverse samples.

Output format: JSON.

Dataset name: {dataset_name}

Column names (in order): {col_names}

Summary statistics:
{summary_stats}

CSV of full data:
{data}
```

[2]LLaMA 3.3-70B failed on *geographical-origin-of-music*, *pumadyn32nh*, *student-performance-por*, *superconductivity*, and *wave-energy* due to token limitations. TabPFN failed on *geographical-origin-of-music* due to extreme dimensionality.

```
Please generate {batch_size} rows of synthetic data.

Treat the rightmost column as the target. Return only a JSON object:
{
  "synthetic_data": "<CSV string>"
}

Do not include any additional text.
```

A.2 SFT-Generation Details

A.2.1 Datasets

1. **Abalone** (OpenML): https://www.openml.org/search?type=data&sort=runs&id=183&status=active
2. **CA Housing** (OpenML): https://www.openml.org/search?type=data&status=active&id=45578&sort=runs
3. **CASP** (OpenML): https://www.openml.org/search?type=data&status=active&id=42903
4. **Diabetes** (OpenML): https://archive.ics.uci.edu/dataset/34/diabetes3
5. **Faults** (UCI): https://archive.ics.uci.edu/dataset/198/steel+plates+faults

A.3 Model Details

We modify the original RealTabFormer implementation to use LLaMA 3.2 (1B, 3B) (Grattafiori et al., 2024), Qwen2.5-3B (Qwen et al., 2025), and Mistral v0.3 7B Jiang et al. (2023). Here, we follow RealTabFormer's base training and sampling hyperparameters in Table 3. To SFT GREAT, we also use its original implementation of which the base hyperparameters can be found in Table 4. For CT-GAN and TVAE, we use the default hyperparameters and implementation found in Synthcity (Qian et al., 2023).

Hyperparameter	Default Value
epochs	1000
batch_size	8
train_size	1
output_max_length	512
early_stopping_patience	5
early_stopping_threshold	0
mask_rate	0
numeric_nparts	1
numeric_precision	4
numeric_max_len	10

Table 3: Numeric hyperparameters of REaLTabFormer.

Hyperparameter	Default Value
epochs	100
batch_size	8
float_precision	None
temperature	0.7
top-k sampling	100
max_length	100

Table 4: GReaT training and sampling hyperparameters.

A.4 Simulated Data Generation Details

For Figure 2b, we initialize a Multivariate Gaussian of N(1e10,1e9). This ensures that there are up to 10 digits for a column as RealTabFormer can struggle to process exceptionally long decimal strings. We then sample

10,000 training and holdout rows for our experiment. At each level, we add additional columns to the training, holdout, and therefore synthetic dataset observation sequence lengths increase by 10 digits for each column.

B Section 5 Experiment details and additional results

B.1 Details of Digit Modifier

Algorithm 1 Digit Modifier Protection Mechanism

Input: real training dataset D, generator G, parameters $p_{\min}, p_{\max}$, probability function g
train G on D
generate synthetic table D_{syn} with entries $x_{k,i}$
let $\mathcal{N}$ be the set of indices of numerical columns; set $X_i = \{x_{1,i}, \ldots, x_{n,i}\}$
for all (k, i, r) with $i \in \mathcal{N}$ and r a digit place considered for $x_{k,i}$ **do**
    $p \leftarrow g(x_{k,i}, X_i; p_{\min}, p_{\max})$
    sample $b \sim \text{Bernoulli}(p)$
    if $b = 1$ **then**
        replace the digit at position r in $x_{k,i}$ with a different digit
    end if
end for

For the digit modifier, we define a probability function

$$g(\mathbf{x_{k,i}}, \mathbf{X}_i; p_{\min}, p_{\max})$$

that assigns a probability to each value $\mathbf{x_{k,i}}$ in column X_i, constrained to lie within the range $[p_{\min}, p_{\max}]$. In our experiment, we set

$$M(\mathbf{X_i}) = \max_{\mathbf{x_{k,i}} \in \mathbf{X_i}} |\mathbf{x_{k,i}}|,$$

which represents the largest absolute value in the numerical column. The probability function is then given by

$$g(\mathbf{x_{k,i}}, \mathbf{X_i}; p_{\min}, p_{\max}) \;=\; p_{\min} + (p_{\max} - p_{\min}) \frac{|\mathbf{x_{k,i}}|}{M(\mathbf{X_i})}.$$

B.2 Details of Tendency-Based logit Processor

Algorithm 2 Tendency-Based Logit Processor Protection Mechanism

Input: real training dataset D, LLM-based generator G, parameter t, and curve-up function f_t
train G on D
while generator G outputs logits at inference time **do**
    scale all logits to $[0, 1]$ using a shifted min–max scaler
    apply the curve-up function f_t to all scaled logits
    rescale the curved-up logits back to the original scale
    apply softmax to obtain the output token distribution
    sample the next token from this distribution, decode it to a digit in the original data space
end while

For the tendency-based logit processor, we need a min–max scaler S_l, a reverse scaler S_l^{-1}, and a curving up function f_t. Specifically, let $l = (l_1, l_2, \ldots l_k)$ be the k logits. Let $m_l = \min_j l_j$, $M_l = \max_j l_j$, and $\varepsilon > 0$ (small constant). Then the min–max scaler S_l is defined componentwise as

$$\big[S_l(l_1, l_2, \ldots, l_n)\big]_i = \frac{l_i - m_l}{M_l - m_l + \varepsilon}, \qquad i = 1, \ldots, k.$$

Similarly, we can also define S_l^{-1} componentwise as

$$\big[S_l^{-1}(s_1, s_2, \ldots, s_n)\big]_i = m_l + s_i\,(M_l - m_l + \varepsilon), \qquad i = 1, \ldots, n,$$

Too see why we need concavity, let's fix $0 < a < b \leq 1$. Since $a = \theta b$ for some $\theta \in (0, 1)$, concavity of f_t implies

$$f_t(a) = f_t(\theta b + (1 - \theta) \cdot 0) \;\geq\; \theta f_t(b) + (1 - \theta) f_t(0) = \theta f_t(b),$$

where we used $f(0) = 0$. Dividing both sides by $a = \theta b$ gives

$$\frac{f_t(a)}{a} \geq \frac{f_t(b)}{b}.$$

Thus, the ratio $f_t(x)/x$ is non-increasing in x. This means smaller logits receive a proportionally larger boost compared to larger logits. Therefore, f_t preserves the ordering of the logits (by monotonicity) while compressing their differences (by concavity), which corresponds to a "curving-up" transformation that favors lower logits.

In particular, the curving function we use in our experiment is $f_t(x) = x^{\frac{1}{t}}$. Figure 5 below demonstrates the graph of our f_t under different t.

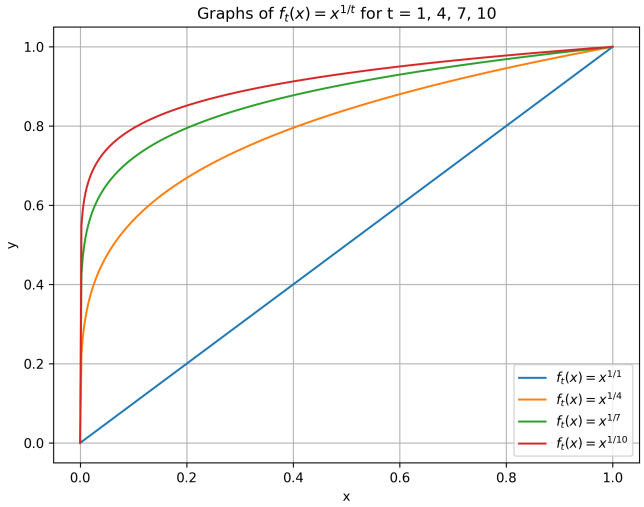
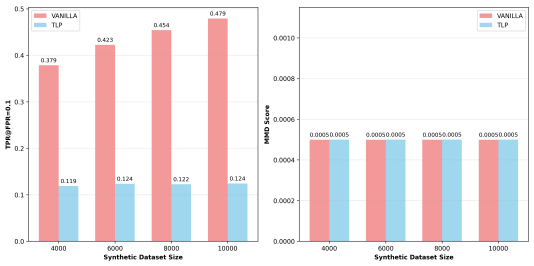


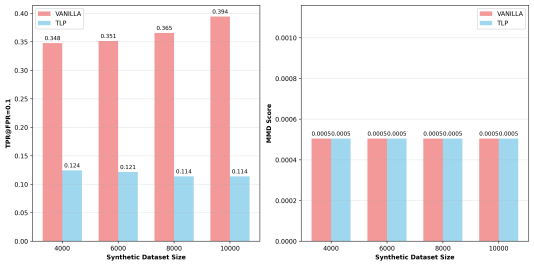
Figure 5: Visualization of the transformation function $f_t(x) = x^{1/t}$ under varying values of t. As t increases, the function becomes more concave, amplifying smaller logits proportionally more than larger ones. This behavior helps compress logit differences while preserving their ordering.

B.3 Tendency-Based Logit Processor Efficacy in Lowering LevAtt True Positive Rate

Please refer to Figure 6 for experimental results showing that TLP lowers the LevAtt TPR to **12.5%** at an FPR of **10%**, while maintaining good fidelity across different synthetic dataset lengths.



(a) Simulation Dataset

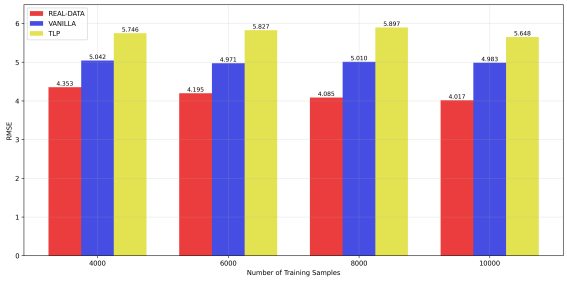


(b) CASP Dataset

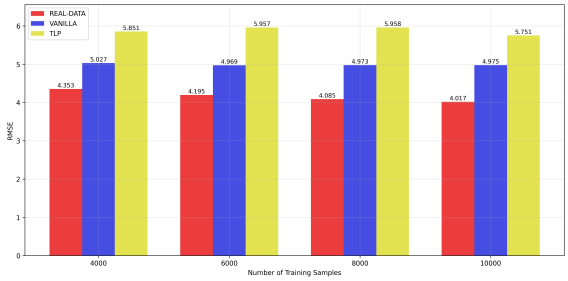
Figure 6: Privacy–utility trade-off of TLP across different datasets using RealTabFormer as the base generative model. Panel (a) reports results on the simulation dataset (5.2), while panel (b) reports results on the CASP dataset. In both cases, TLP consistently reduces the LevAtt TPR to about 12.5% at 10% FPR across all synthetic dataset sizes, indicating a substantial improvement in privacy protection compared to the vanilla inference procedure. Here, the vanilla synthetic dataset is generated using the default RealTabFormer inference without any protection mechanism. Meanwhile, the Maximum Mean Discrepancy (MMD), computed between the training data and the synthetic data generated by either vanilla or TLP, remains nearly unchanged, demonstrating that TLP preserves fidelity while mitigating privacy leakage.

B.4 TENDENCY-BASED LOGIT PROCESSOR PRIVACY VS MACHINE LEARNING EFFICACY

We refer for to Figure 7 for a full description of this experiment.



(a) CASP dataset:Utility comparison of XGBoost models trained on real data, vanilla synthetic data, and TLP-protected synthetic data, demonstrating that TLP reduces the LevAtt AUC-ROC to below 55%.



(b) CASP dataset: Utility comparison of XGBoost models trained on real data, vanilla synthetic data, and TLP-protected synthetic data, demonstrating that TLP reduces the LevAtt TPR to below 12.5% at FPR $= 10\%$.

Figure 7: Utility comparison of XGBoost models trained on real, vanilla synthetic, and TLP-protected synthetic data. Models trained on real data achieve the lowest RMSE, while vanilla synthetic data incurs a moderate increase and TLP-protected data shows a larger degradation. Importantly, the performance gap between vanilla and TLP remains relatively stable as training size increases. Even with stronger tendency parameters required for larger synthetic datasets to meet privacy thresholds, the utility degradation appears bounded, highlighting the desirable property of TLP in controlling the privacy–utility trade-off.

C STATEMENT OF LLM USE

We used LLMs to assist in the programming of our experiments, designing data visualizations, making the codebase more user-friendly, finding clearer phrasings in our writing, and formatting Latex code. All additions by LLMs were checked by the authors.