Title

Membership Inference Attacks Against Tabular Synthetic Data Under Conservative Threat Models

ISBN

9798247901075

Author

Ward, Joshua

Publication Date

2026-05-21

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA

Los Angeles

Membership Inference Attacks Against Tabular Synthetic Data Under Conservative Threat Models

A dissertation submitted in partial satisfaction

of the requirements for the degree

Doctor of Philosophy in Statistics

by

Joshua John Ward

2026

ABSTRACT OF THE DISSERTATION

Membership Inference Attacks Against Tabular Synthetic Data
Under Conservative Threat Models

by

Joshua John Ward

Doctor of Philosophy in Statistics
University of California, Los Angeles, 2026
Professor Guang Cheng, Chair

Tabular synthetic data has emerged as a promising mechanism for privacy-preserving data sharing, enabling organizations in sensitive domains such as healthcare and finance to publish data products that preserve statistical utility while limiting risk to individuals in their training sets. The empirical privacy of synthetic data releases is most rigorously assessed through Membership Inference Attacks (MIAs). However, existing MIAs for tabular synthetic data share two critical limitations: they frequently assume unrealistic adversarial knowledge of the generative model's implementation that would be withheld under responsible release practices, and they operate exclusively over the Cartesian feature space of a single observation, failing to capture vulnerabilities introduced by Large Language Models and relational data generators whose inductive biases differ fundamentally from earlier generative models.

This dissertation addresses both limitations through three contributions. The first introduces the Generative Likelihood Ratio Attack (Gen-LRA), a No-Box MIA grounded in an influence function framework, accompanied by theoretical results characterizing what the attack measures and when it succeeds. Gen-LRA achieves state-of-the-art performance across a benchmark spanning 35 datasets, 9 generative architectures, and over 1,500 synthetic dataset configurations.

The second contribution studies privacy risks specific to Large Language Model (LLM)-

based tabular generators, which encode tabular rows as strings and generate records via autoregressive text production—a paradigm invisible to feature-space attacks. We introduce LevAtt, a No-Box MIA that targets memorized sequences of numeric digits in synthetic string representations, achieving near-perfect membership classification on some state-of-the-art generators.

The third contribution formalizes membership inference for the multi-table synthetic data setting, where a user's information is distributed across multiple interconnected tables in a relational database. We show that single-table MIAs systematically underestimate user-level privacy leakage in this setting, and introduce MT-MIA, a No-Box attack that represents each user as a heterogeneous subgraph and leverages Graph Neural Networks to perform user-level membership inference.

Together, these three works advance a principled framework for realistic, architecture-aware privacy auditing of tabular synthetic data.

The dissertation of Joshua John Ward is approved.

Yuan Tian

Yingnian Wu

Suhas Diggavi

Guang Cheng, Committee Chair

University of California, Los Angeles

2026

I became interested in tabular synthetic data after listening to a talk by the then new-to-the-department Professor Guang Cheng at the end of the Spring Quarter of 2022. I vividly remember in our first meeting Guang said, "I will take on anyone as a student, my only expectation is that you work very hard. If you take me on as an advisor, your expectation should be you will have a hard time for a year or two as you find a topic, before you then hit your stride." I thought that I would find a topic quickly, after all, synthetic data and trustworthy AI were a new field. To my dismay, Guang was right and the first two years of my doctorate were a frustrating ordeal, characterized by the stress of switching topics several times and being immensely stressed about my productivity from an output perspective.

I have an enormous amount of gratitude, appreciation, and respect for Guang for sticking with me and supporting me through what was in hindsight a dark but important time. His support in terms of his flexibility, as well as his patience in allowing me to wander the wilderness of learning the practice of being an independent researcher were a tremendous gift. To his credit, Guang was right and in the final two years, I found a topic which I love and my rhythm which led to a series of papers, some of which make up this dissertation.

From this journey, I also want to thank Chi-Hua Wang who is a co-author on much of this work. His partnership was invaluable as a thought partner but also as a mentor on how to do good science. I also want to thank the many folks I've collaborated with, the graduate students and post-docs of the UCLA Trustworthy AI lab as well as the broader Statistics department of students, professors and staff, whose presentations, work and camaraderie were vital to the success of this work and my growth as a researcher.

I want to thank my family and friends for bearing with me. My experience is that working on Statistics research is a wholly isolating experience, as very few people can understand what you are doing and thinking about all day. Your love and wanting to be around me did more for this success than any grant or opportunity ever could.

Lastly, I want to thank my fiance Megan, who has been with me since before I even started my Master's in 2020. Your unwavering confidence, empathy and commitment to us is the hidden PI that kept me on track and pushed me to finish this. Long nights writing and rambling about buggy code at dinner do not always make for a pleasant partner. I love you so much.

Contents

List of Figures

List of Tables

Acknowledgements

Chapter 2 is a version of Ward, J., Wang, C., & Cheng, G. "Privacy Auditing Synthetic Data Release through Local Likelihood Attacks." Under review at the ACM Conference on Computer and Communications Security (CCS). Available at https://arxiv.org/abs/2508.21146. As first author, I led the conception of the project, designed and implemented the local likelihood attacks, conducted all experiments, and wrote the manuscript. C. Wang, a postdoctoral researcher, contributed to co-ideation of the project and editing of the manuscript. G. Cheng, as advisor and PI directing this research, provided guidance on research direction, methodology, and manuscript revision.

Chapter 3 is a version of Ward, J., Wang, C., & Cheng, G. "Multi-Table Membership Inference Attacks for Synthetic Relational Data Release." Under review at the ACM Conference on Computer and Communications Security (CCS). Available at https://arxiv.org/abs/2602.07126. As first author, I led the conception of the project, designed and implemented the multi-table membership inference attacks, conducted all experiments, and wrote the manuscript. C. Wang, a postdoctoral researcher, contributed to co-ideation of the project and editing of the manuscript. G. Cheng, as advisor and PI directing this research, provided guidance on research direction, methodology, and manuscript revision.

Chapter 4 is a version of Ward, J., Gu, B., Wang, C., & Cheng, G. (2026). "When Tables Leak: Attacking String Memorization in LLM-Based Tabular Data Generation." *Proceedings on Privacy Enhancing Technologies (PETs)*, 2026. Available at https://arxiv.org/abs/2512.08875. As first author, I led the conception of the project, designed and implemented the attack methodology, conducted all experiments, analyzed the results, and wrote the manuscript. B. Gu contributed to experiments, coding, and writing of the defenses section of the paper. C. Wang, a postdoctoral researcher, contributed to co-ideation of the project and editing of the manuscript. G. Cheng, as advisor and PI directing this research, provided guidance on research direction, methodology, and manuscript revision.

This work was financially supported by the National Science Foundation under award NSF-CNS 2247795, the Office of Naval Research under award ONR N00014-22-1-2680, a gift from Cisco, and the National Artificial Intelligence Research Resource Pilot (NAIRR-250187).

Vita

Education

Whitman College — Fall 2014 - December 2017

Bachelor of Arts, Economics and Mathematics

University of California, Los Angeles — September 2020 - June 2022

Master of Science, Applied Statistics

Academic Appointments

Food and Drug Administration — Nov 2023 - May 2025

ORISE AI Research Fellow

Arizona State University, Learning Engineering Institute — Oct 2024 - May 2025

AI Researcher

Industry Experience

Idem Labs — February 2026 - Present

CTO / Cofounder

Amazon — June 2023 - September 2026

AI Applied Science Intern, Amazon Search

Accenture — June 2021 - December 2021

AI Research Intern, R&D

Honors and Awards

UCLA Master's of Applied Statistics Fellowship — 2020

UC Native American Opportunity Program — 2022

FDA ORISE AI Fellowship — 2023

NAIRR Pilot Award (250187) — 2024

Best Paper Award, *Clinical and Translational Science* — 2025

Selected Publications

Ward, J., Gu, B., Wang, C., & Cheng, G. When Tables Leak: Attacking String Memorization in LLM-Based Tabular Data Generation. *Proceedings on Privacy Enhancing Technologies (PETs)*, 2026.

Ward, J., Yang, C., Wang, C., & Cheng, G. Ensembling Membership Inference Attacks Against Tabular Generative Models. *Workshop on Artificial Intelligence and Security*, 2025.

Zeng, X., Ward, J., & Cheng, G. FairRR: Pre-processing Group Fairness Through Randomized Response. *AISTATS*, 2024.

Weng, C., Ward, J., Lin, W., Dong, S., Liu, Q., & Zhang, H. Out-of-distribution detection as a risk-control strategy for medical classification machine learning models. *Clinical and Translational Science*, 2025. (Best Paper)

Ward, J., Werner, M., Savran, W., & Schoenberg, F. Evaluation of ETAS and STEP forecasting models for California seismicity using point process residuals. *Environmetrics*, 36, e70014, 2025.

Ward, J., Basu, T., Menzer, O., & SenGupta, I. A novel implementation of Siamese-type neural networks in predicting rare fluctuations in financial time series. *Risks*, 10(2), 39, 2022.

Chapter 1

Introduction

Real-world tabular data is often privacy-sensitive to the individual observations that compose these datasets, hindering their ability to be shared in open-science efforts that can aid in new research and improve reproducibility. Sensitive tabular data is prevalent across consequential domains—from electronic health records and insurance claims to financial transactions and government surveys—where the data holder is often legally or ethically constrained from sharing original records. A promise of generative modeling is that models trained on sensitive data can produce synthetic samples that preserve the privacy of the training set while maintaining much of its intrinsic statistical information, enabling responsible release to a third party [1–4]. These benefits are especially valuable in privacy-sensitive sectors such as medicine [5] and banking [6], where regulatory constraints often limit access to original data, and as a result synthetic data generation has emerged as an essential technique for expanding machine learning capabilities in data-constrained and privacy-regulated environments [7, 8].

To audit the empirical privacy of synthetic data generators, membership inference attacks (MIAs) have recently been extended from traditional machine learning models to synthetic tabular data. Here, privacy auditing is framed as an adversarial game: given specific constraints defined by a threat model, an attacker attempts to determine whether a test observation belongs to a model's training dataset, exploiting some notion of model failure [9–11]. A successful attack represents a concrete privacy breach with clear real-world implications, where other similarity-based metrics have been shown to fail to capture privacy risk [7, 12, 13].

While promising, MIAs for generative models and synthetic data release have seen limited success for two reasons. First, the most powerful existing attacks—shadow-box and query-based methods—require the adversary to know the implementation of the target generative model in order to train surrogate copies [14–16]. In practice, an organization following responsible release practices will not disclose its model architecture

or code [17], trivially defeating such attacks. These attacks also do not scale computationally to modern generators, where a single training run can take hours. The No-Box threat model—in which the adversary observes only the released synthetic data and a reference sample from the population—is the realistic setting for synthetic data release, yet prior No-Box attacks have lacked principled theoretical foundations and have been outperformed on benchmarks by less realistic alternatives.

Second, existing MIAs operate exclusively over the Cartesian feature space of a single tabular observation. This design was appropriate for early generators such as GANs and VAEs, which model distributions directly over tabular features. However, modern tabular generative architectures introduce inductive biases that create privacy leakage through channels orthogonal to the feature space, and feature-space attacks are blind to these vulnerabilities by construction. Large Language Models (LLMs) encode tabular rows as strings and generate new records via autoregressive text production, introducing a vulnerability to digit-level memorization that no feature-space attack can detect. In the relational setting, where a user's information is distributed across items in multiple interconnected tables in a relational database, the unit of privacy is not a single row but an entire user entity spanning multiple tables—a structure that single-table, item-level MIAs are fundamentally unable to target.

This dissertation addresses both of these limitations. Each chapter presents a membership inference attack that operates under a realistic No-Box threat model and is designed to target the specific inductive biases of a class of tabular generative models. Together, these three works establish that effective privacy auditing of tabular synthetic data requires both conservative adversarial assumptions and sensitivity to the architectural properties of the generator being audited.

Chapter 2: Privacy Auditing Synthetic Data Release through Local Likelihood Attacks. We introduce the Generative Likelihood Ratio Attack (Gen-LRA), a No-Box MIA for tabular synthetic data that targets privacy leakage arising from local overfitting in a principled way. We focus on the No-Box threat model because it is the most realistic setting for synthetic data release: we make no adversarial assumptions about model architecture, access, or training parameters, mimicking real-world scenarios of parties following best practices. Under this threat model, Gen-LRA constructs an influence function formulated from likelihood ratio estimation to detect when a test point's inclusion shifts the estimated likelihood of the synthetic data in its local neighborhood. We develop three theoretical results supporting Gen-LRA: an invariance property showing that the log-likelihood-ratio score is insensitive to invertible reparametrizations of the data; a closed-form characterization showing that Gen-LRA computes a localized density-ratio statistic; and a mean-score-gap result that, under a general model of local overfitting, produces a provable separation between members and non-members. We validate these predictions in a controlled

simulation study and demonstrate empirical dominance across a benchmark of 35 datasets, 9 generative architectures, and 9 competing attacks—over 1,500 unique synthetic dataset configurations—where Gen-LRA is the top-ranked attack on more than twice as many configurations as any baseline.

Chapter 3: Membership Inference Attacks for the Multi-Table Synthetic Data Setting. We formalize, to our knowledge the first work to do so, privacy auditing for multi-table synthetic data generators, which produce synthetic relational databases rather than single tables. While considerable progress in synthetic data generation has focused on single-table applications, most real-world data exists in relational databases where a user's information is distributed across items in multiple interconnected tables. We show that applying existing item-level MIAs to individual tables in a synthetic database underestimates user-level privacy leakage because privacy leakage of a particular item implies the leakage of all connected items in the user's subgraph. We then propose Multi-Table MIA (MT-MIA), a No-Box attack that constructs relational databases as heterogeneous graphs, infers whether a test subgraph describing all connected information for a user was in the training set, and uses Heterogeneous Graph Neural Networks (HGNNs) to learn representations of user-centric subgraphs for membership classification. We construct examples in which current single-table, item-level attacks perform at random chance while MT-MIA achieves near-perfect AUC, and find that current state-of-the-art multi-table generators possess this unique vulnerability on real-world datasets even under our conservative threat model.

Chapter 4: When Tables Leak: Attacking String Memorization in LLM-Based Tabular Data Generation. We study privacy risks unique to LLM-based tabular generators, which generate data in an intermediate string space that is invisible to feature-space attacks. LLMs have been demonstrated to exhibit strong tendencies to memorize training data, particularly structured and repeated patterns [18, 19]. We show that in the tabular setting, this tendency manifests in the memorization of numeric digit sequences— the structured, often repeated digit patterns of fields such as account numbers, ZIP codes, and dates that appear in tabular training data. To measure this risk, we introduce LevAtt, a No-Box MIA that uses Levenshtein distance on string-encoded synthetic observations to detect memorized digit patterns. Even under a conservative threat model with access only to the synthetic data, LevAtt exposes substantial privacy leakage across both fine-tuned and in-context LLM generators, achieving near-perfect membership classification on some state-of-the-art models and exposing leakage that conventional feature-space MIAs fail to detect entirely. We also propose two defenses: a post-processing Digit Modifier and a novel Tendency-based Logit Processor (TLP) that intervenes during sampling to perturb digit generation, finding that TLP can effectively control digit memorization with minimal loss of statistical fidelity or downstream utility.

Chapter 2

Privacy Auditing Synthetic Data Release through Local Likelihood Attacks

2.1 Introduction

Real-world tabular data is often privacy-sensitive to the individual observations that compose these samples, hindering their ability to be shared in open-science efforts that can aid in new research and improve reproducibility. A promise of generative modeling is that models trained on sensitive data can produce samples that preserve the privacy of the training set while maintaining much of its intrinsic statistical information, enabling responsible release to a third party. In practice, a wide array of methodologies have been proposed to accomplish synthetic data release involving modifying loss functions [20, 21], creating new generative model architectures [1, 22], and studying data release strategies [23–25] to provide differential privacy guarantee. In another direction, a variety of methods have been proposed that maximize the fidelity of synthetic data and argue that privacy is satisfied through ad-hoc similarity metrics [26–29].

To audit the empirical privacy of synthetic data generators, membership inference attacks (MIAs) have recently been extended from traditional machine learning models to synthetic tabular data. Here, privacy auditing is framed as an adversarial game: given specific constraints defined by a threat model, an attacker attempts to determine whether a test observation belongs to a model's training dataset exploiting some

The contents of this chapter appeared in: Ward, J. et al., "Privacy Auditing Synthetic Data Release through Local Likelihood Attacks," 2026.

notion of model failure [9–11]. A successful attack represents a concrete privacy breach with clear real-world implications, where other similarity-based metrics have been shown to fail to capture privacy risk [7, 12, 13].

While promising, MIAs for generative models and synthetic data release have seen limited success. Previous work has often relied on distance- or density-based heuristics, or has made additional assumptions about model query access that are unrealistic to the release setting and do not scale computationally to modern architectures. Moreover, most existing attacks are proposed without a theoretical account of what they measure or when they should succeed, making it difficult to assess whether empirical performance reflects a principled design or a coincidence of benchmark choice. We address both gaps in this work.

We focus on studying membership inference for the release of synthetic data in a No-Box Threat Model [15]. In this setting, we make no adversarial assumptions of knowledge about model architecture, access, or training parameters, mimicking real-world scenarios of parties following best practices for releasing synthetic data in domains like healthcare and finance. Under this threat model, we derive a powerful MIA called *Generative Likelihood Ratio Attack* (Gen-LRA), which constructs an influence function formulated from likelihood ratio estimation to target privacy leakage that occurs through model overfitting. We develop a theoretical framework that characterizes what Gen-LRA computes and predicts when it succeeds, validate these predictions in a controlled simulation study, and demonstrate empirical dominance across a large-scale benchmark of 1,525 synthetic dataset configurations.

Contributions:

1. *A novel attack.* We introduce Gen-LRA, a No-Box MIA that formulates membership inference as the influence of a test point on a surrogate model's estimate of the likelihood ratio over a local subset of synthetic data. To our knowledge, this is the first MIA for tabular synthetic data to explicitly adapt an influence-function framework to the No-Box setting.

2. *Theoretical framework.* We develop three theoretical results supporting Gen-LRA: (i) an invariance property showing that the log-likelihood-ratio score is insensitive to invertible reparametrizations of the data; (ii) a closed-form characterization showing that Gen-LRA is, to leading order, a localized density-ratio statistic, making explicit what the attack measures and why it differs from prior density-based MIAs; and (iii) a mean-score-gap result that, under a general model of local overfitting, produces a provable separation between members and non-members and identifies the structural quantities that govern the attack's success.

3. *Simulation validation.* We employ a controlled simulation that directly instantiates the overfitting model, validating our theoretical results and characterizing when membership inference is and is not feasible. We find Gen-LRA is uniquely tunable across overfitting regimes, and the attack outperforms

existing methods particularly when a generator approximately memorizes its training data.

4. *Empirical benchmark.* We show that Gen-LRA broadly outperforms competing MIAs across a benchmark of 35 datasets, 9 generative architectures, and 9 attacks, comprising 1,525 unique synthetic dataset configurations and over 10,000 attack runs. Gen-LRA is the top-ranked attack on more than 2× as many runs as any baseline across both AUC and TPR at all fixed false positive rates (Table 2.1), and achieves the highest mean values across the top 100 highest-leakage runs on every metric (Table 2.2). We additionally perform ablations across encoding strategies, surrogate density estimators, and locality and bandwidth choices, finding that Gen-LRA is robust across these design decisions.

2.2 Membership Inference Attacks Preliminaries

In this work, we study the Membership Inference Attack Game in the context of synthetic data release. The objective of this game is to determine whether a particular data point was included in the original training dataset by examining the outputs of a generative model. We first introduce the formal definition of the Membership Inference Attack Game modeled after [9].

2.2.1 Membership Inference Attack Game

Definition (Membership Inference Attack Game) The game proceeds between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$ as follows:

1. The challenger samples a training dataset $T = \{x_i\}_{i=1}^{n}$ from the population distribution $x_i \sim \mathbb{P}$ over the domain $\mathcal{X} = \mathcal{X}_1 \times \cdots \times \mathcal{X}_d$, where each attribute domain $\mathcal{X}_j$ is either numeric (i.e., $\mathcal{X}_j \subseteq \mathbb{R}$) or categorical (i.e., $\mathcal{X}_j$ is a finite discrete set), and uses T to train a tabular generative model $G \leftarrow \mathcal{T}(T)$. The generative model G produces synthetic dataset S.

2. The challenger flips a bit $b \in \{0, 1\}$. If $b = 0$, the challenger samples a test observation $x^\star \in \mathcal{X}$ from the population distribution $\mathbb{P}$. Otherwise, the challenger selects the test observation $x^\star$ from the training set T.

3. The challenger sends the test observation $x^\star$ to the adversary $\mathcal{A}$.

4. The adversary has access to some information defined by a threat model and uses this information to output a guess $\hat{b} \leftarrow \mathcal{A}(x^\star)$.

5. The output of the game is 1 if $\hat{b} = b$, and 0 otherwise. The adversary wins if $\hat{b} = b$, i.e., if it correctly identifies whether $x^\star$ was part of the training set T or a sampled point from $\mathbb{P}$.

Adversary's goal and capabilities. The adversary $\mathcal{A}$ aims to determine whether a specific data point $x^\star$ was part of the original training dataset T or was drawn from the population distribution $\mathbb{P}$. The adversary can use any available information to construct a scoring function that classifies the membership of $x^\star$ defined by a threat model. The performance of this classifier, evaluated with binary classification metrics, measures the privacy leakage of the training data for G through S. Formally, a membership inference attack can be expressed as

$$\mathcal{A}(x^\star) \;=\; \mathbb{I}[f(x^\star) > \gamma]\,, \tag{2.1}$$

where $\mathbb{I}$ is the indicator function, $f(x^\star)$ is a scoring function of $x^\star$, and γ is an adjustable decision threshold.

2.2.2 Threat Model

In this work, we consider a "No-Box" [15] threat model in which the adversary has no access to the internal structure, parameters, or sampling mechanism of the generative model. The attack must instead be constructed using only two observed datasets: the released synthetic dataset $S \sim G(T)$, and an independently collected reference dataset $R \sim \mathbb{P}$ drawn from the same underlying population. The adversary is not granted access to knowledge of the model implementation nor can they issue queries to a trained generator. This reflects deployment scenarios in which organizations release synthetic data for downstream analysis while keeping all model knowledge confidential. The synthetic dataset S serves as the only potential leakage surface, and the reference set R provides a statistical anchor for the population. A reference dataset is commonly assumed in No-Box attacks on synthetic data [10, 15, 30, 13] as well as in MIAs for supervised learning models [11, 31, 32], and represents a scenario in which an adversary may be able to find comparable data in the real world—for example via open-source datasets, paid collection, or domain knowledge.

Three considerations motivate our focus on this threat model. First, it is the most realistic setting for synthetic data release. Shadow-box attacks [14–16] rely on the adversary knowing the generator's architecture or implementation in order to train surrogate models, but a defender following release best practices can trivially neutralize such attacks by simply withholding this information [17]. The estimated privacy leakage of these attacks is therefore not conditioned on the knowledge an adversary would realistically have. Second, shadow-box and white-box approaches do not scale computationally to modern tabular generators. Diffusion-based and transformer-based architectures can take hours to train for a single model, and attacks that require fitting hundreds of surrogate models per auditing run become infeasible as the generator's training cost grows. Third, No-Box attacks study leakage through the synthetic data itself rather than through any particular model, making them model-agnostic. This property makes them useful both as a practical privacy-auditing tool that a third party can run on released synthetic data, and as a research tool for understanding the

privacy behavior of generative models independent of their specific architectures.

2.3 Related Works

2.3.1 Assessing Overfitting in Tabular Generative Models

Several measures have been developed to assess the fitness of tabular synthetic data, particularly from a privacy perspective. These metrics generally aim to measure the similarity between the training and synthetic datasets, with the ideal outcome being that the synthetic data is neither too similar to the training data nor too different. A widely used metric for this purpose is the Distance to Closest Record[1] [33, 28, 34, 35, 26, 27], which measures the distance from each training point to its nearest neighbor in the synthetic dataset and averages across training points. Another commonly used metric is the Identical Matching Score (IMS) [34, 36, 37], which measures the proportion of identical records between the training and synthetic datasets. While these measures can be useful for describing overfitness from a distribution-level quality or model-generalization perspective, they do not characterize privacy risk: there is no assumed threat model and they are not evaluated against non-member examples.

Exploiting overfitting as a source of privacy leakage has been documented in specific generators. [30] showed that TVAE [38] overfits to minority-class examples in a medical training dataset, leaking their privacy. [13] found that Tab-DDPM [3] heavily replicates training records from certain demographic subgroups when generating synthetic data for the Adult dataset. These findings motivate our approach of designing an attack that explicitly targets the local overfitting behavior of tabular generative models (see Section 2.4).

2.3.2 MIAs for Tabular Generative Models

Membership inference attacks explicitly characterize the empirical privacy risk of a machine learning model [39, 40]. MIAs were originally developed for attacking supervised learning classifiers [9], where the general idea is to query a model with different observations to learn patterns in its class-probability outputs. Membership is then inferred by comparing the outputs of the model to outputs from reference models in various ways [11, 41, 32, 42, 31, 43].

To adapt to the structural differences of generative models, a range of MIAs for tabular generators have been proposed that employ different threat models and strategies [10, 44, 45, 15, 16, 14, 30, 13]. Gen-LRA

[1]DCR in the similarity-metric case compares a training point to a synthetic point. [10] propose an MIA in which the scoring function is a distance computation for a test point and a synthetic point; in all other sections of the paper we use DCR to refer to the MIA.

is most closely related to DOMIAS [30] and to a line of work that extends query-based attacks to tabular generative models [15, 16, 14].

Relation to DOMIAS DOMIAS follows the same threat-model assumptions as Gen-LRA and similarly defines its scoring function in (2.1) as a density ratio, $\frac{p_S(x^*)}{p_R(x^*)}$. However, its theoretical foundation is limited because it effectively tests the wrong membership hypothesis: rather than testing whether the inclusion of x^* in the reference data influences the released synthetic dataset, it tests only whether x^* lies in a region where the synthetic distribution is denser than the reference distribution. In contrast, Gen-LRA directly targets the membership inference problem by measuring the effect of specifically including x^* on R, angling with the true hypothesis of interest. Moreover, while DOMIAS relies on a single-point density estimate, Gen-LRA aggregates evidence over a local region, allowing it to capture diffuse memorization signals and incorporate substantially more information. Our theoretical analysis, simulations, and empirical benchmarks all support this distinction: because DOMIAS tests the wrong hypothesis, its performance degrades substantially in a variety of overfitting regimes, whereas Gen-LRA remains robust and consistently outperforms it.

Relation to query-based attacks [14], [15], and [16] propose query-based attacks on tabular generators that additionally assume the adversary has knowledge of the *implementation* of the target model. In these methods, an attacker trains many versions of the model on $R \cup \{x^\star\}$ and R, generates many synthetic datasets, represents each synthetic dataset by summary statistics or histograms, and trains a classifier to distinguish between the two regimes.

Gen-LRA improves on these attacks in two ways. First, they are unsuitable for auditing privacy in a release setting because they are trivially defeated if the defender chooses not to disclose the architecture; indeed, [17] has shown significant privacy leakage arising from architectural disclosure, leading to the recommendation that data-releasing parties disclose as little model information as possible. Gen-LRA makes no assumption about model implementation. Second, query-based attacks are computationally expensive: they require training $(N_{\text{test}} + 1) \times N_{\text{surrogate}}$ separate models to audit a single generator, which is impractical as large diffusion and language model architectures become more common. Gen-LRA requires only $N_{\text{test}} + 1$ density estimators to be fit, which is substantially cheaper.

2.4 Generative Likelihood Ratio Attack

In this section, we propose *Generative Likelihood Ratio Attack* (Gen-LRA), a membership inference attack designed to detect membership leakage in synthetic data through a statistical notion of *likelihood influence*.

Unlike prior MIAs that evaluate the density or distance of a test point itself, Gen-LRA frames membership inference as a measurement of how much the test point $x^\star$ influences an estimate of the likelihood of S. The central idea is that if $x^\star$ belonged to the training set and the generative model is overfit, then adding $x^\star$ to a surrogate density estimator should increase the estimated likelihood of the synthetic dataset.

We develop Gen-LRA in six steps. Section 2.4.1 reviews empirical influence functions, which provide the theoretical foundation for the attack. Section 2.4.2 defines the Gen-LRA score as a log-likelihood-ratio influence function. Section 2.4.3 justifies the log-likelihood-ratio form via Neyman-Pearson optimality and an invariance property. Sections 2.4.4 and 2.4.5 establish the two main theoretical results of the paper: a closed-form characterization of the Gen-LRA score, and a mean-score-gap result quantifying the attack's power under a general overfitting model. Section 2.4.6 describes the practical implementation, with each design choice motivated by the preceding theory.

2.4.1 Empirical Influence Functions

Influence functions, originally developed in the field of Robust Statistics [46, 47], measure how statistical estimates change when the underlying data distribution is perturbed. The influence function for an estimator θ applied to a distribution F is

$$\mathcal{I}(x^\star, F, \theta) \;=\; \lim_{\epsilon \to 0} \frac{\theta((1-\epsilon)F + \epsilon\delta_{x^\star}) - \theta(F)}{\epsilon}, \tag{2.2}$$

where $\delta_{x^\star}$ is the Dirac measure placing unit mass at $x^\star$. This quantity captures the sensitivity of θ to infinitesimal perturbations of F at $x^\star$.

In the empirical setting with finite samples $\mathcal{D} = \{x_1, \ldots, x_n\}$, the empirical influence function is defined by evaluating how an estimate changes when a point is added:

$$\hat{\mathcal{I}}(x^\star, \mathcal{D}, \theta) \;=\; \theta(\mathcal{D} \cup \{x^\star\}) - \theta(\mathcal{D}). \tag{2.3}$$

For supervised learning models, this difference is typically measured on loss or empirical risk [48]. In an MIA for tabular generative models with no model access, however, measures of loss are not available. Rather than examining how $x^\star$ affects model parameters, Gen-LRA considers the influence of $x^\star$ on the likelihood assigned to generated samples S by a surrogate estimator.

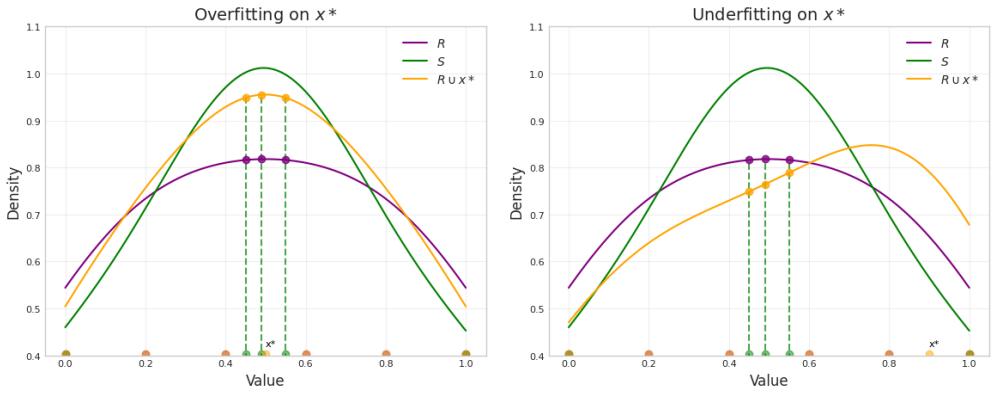


Figure 2.1: A geometric intuition for Gen-LRA with a 1-dimensional toy example. We visualize the KDE plots of $R, R \cup \{x^\star\}, S$ as well as the estimated densities of the synthetic observations over R and $R \cup \{x^\star\}$. Left: $x^\star = 0.5$. The likelihood of the synthetic observations (product of orange intersections) is higher under the density estimate of $R \cup \{x^\star\}$ than R (product of purple intersections), so the attack concludes $x^\star \in T$. Right: $x^\star = 0.9$; the opposite holds, so $x^\star \notin T$ is concluded.

2.4.2 Likelihood Influence as an Attack Surface

Recalling that $T, R \sim \mathbb{P}$ and that our goal is to infer whether $x^\star \in T$ given S and R, we hypothesize that if the generator is overfit [2] then S carries signal of $x^\star$ specifically when $x^\star \in T$. Gen-LRA measures this overfitness by formalizing an influence function on the estimated log-likelihood of S under two surrogate models: one fit on the reference dataset R, and another fit on the augmented dataset $R \cup \{x^\star\}$:

$$\hat{\mathcal{I}}(x^\star; R, S) \;:=\; \log \hat{p}(S \mid R \cup \{x^\star\}) - \log \hat{p}(S \mid R). \tag{2.4}$$

Intuitively (see Figure 2.1), if adding $x^\star$ to R substantially increases the estimated likelihood of S, this suggests $x^\star$ contributed to the generative process. If the likelihood is unchanged or decreases, it implies $x^\star \notin T$. Using $\hat{\mathcal{I}}(x^\star; R, S)$ as the scoring function f in the membership prediction rule (2.1) defines the Gen-LRA attack.

2.4.3 Justification of the Log-Likelihood Ratio

The influence function in (2.4) is one of many possible functionals of $\hat{p}(S \mid R \cup \{x^\star\})$ and $\hat{p}(S \mid R)$. Two properties, one statistical and one geometric, justify the choice of the log-likelihood ratio.

[2] We use *overfitting* for the generator's behavior of placing excess mass near training points relative to the true population, and *memorization* for its geometric manifestation in the synthetic data. Both notions are formalized in Section 2.4.5.

Neyman-Pearson optimality The membership inference game is fundamentally a binary hypothesis test: $H_0 : x^\star \sim P$ against $H_1 : x^\star \in T$. By the Neyman-Pearson lemma, the likelihood ratio test is uniformly most powerful among all tests at a given false positive rate. This principle has long been central to the MIA literature: [9] frame membership inference as a classification problem whose Bayes-optimal solution reduces to a likelihood ratio, [32] derive the Bayes-optimal MIA strategy and show it takes a likelihood-ratio form, and [31] argue for calibrated likelihood-ratio statistics as a general design principle.

[11] make this argument operational, demonstrating that MIAs grounded explicitly in likelihood ratios (rather than ad-hoc distance, confidence, or loss statistics) achieve substantially stronger guarantees in the supervised-learning setting, particularly at the low false positive rates that matter most for practical privacy auditing. Subsequent work has extended this principle to enhanced and low-cost variants [31, 43]. Gen-LRA adapts this design principle to the No-Box tabular synthetic data setting: the scoring function in (2.4) is a log-likelihood ratio over S under two hypotheses about the data distribution that generated it, making it the natural Neyman-Pearson analogue for the attack surface available to a No-Box adversary.

Invariance to reparametrization A second advantage of the log-likelihood-ratio form is that it is invariant to invertible transformations of the data.

Theorem 1 (Invariance of the population influence function)**.** *Let S and R be sets of samples and $x^\star$ a test point, with probability distributions on $\mathcal{X} \subseteq \mathbb{R}^d$. For any continuously differentiable invertible function $g : \mathcal{X} \to \mathcal{X}$ with non-vanishing Jacobian, the population-level log-likelihood-ratio influence function is invariant:*

$$\mathcal{I}(g(x^\star), g(R), g(S)) \;=\; \mathcal{I}(x^\star, R, S). \tag{2.5}$$

The proof (Appendix 2.A.1) follows from the change-of-variables formula and cancellation of Jacobian terms. This invariance is practically important: tabular data is typically preprocessed through one-hot encoding, scaling, or learned embeddings, and an attack whose score depends on these arbitrary choices would be unreliable. Competing attacks for example based on distance heuristics such as Distance to Closest Record do not satisfy this property.

2.4.4 Closed-Form Characterization of the Attack Score

Having justified the log-likelihood-ratio form, we now analyze what Gen-LRA computes. To make this analysis concrete, we specialize to a particular choice of surrogate density estimator and localization strategy. Throughout the remainder of this section, we take $\hat{p}_R$ to be a Gaussian Kernel Density Estimator (KDE) fit

on the reference set R, and we localize the attack statistic to the k-nearest synthetic neighbors of $x^\star$ rather than summing over all of S.

Both of these choices are justified empirically and discussed in full in Section 2.4.6; briefly, Gaussian KDEs admit a closed-form expansion that makes the subsequent analysis tractable and empirically outperform more expressive density estimators (Section 2.7.3), while localization concentrates the statistic on the region where overfitting signal can exist.

Formally, let $\hat{p}_R$ denote a Gaussian KDE fit on reference set R with $|R| = m$ and bandwidth h, let $K_h(u) = h^{-d}K(u/h)$ be the Gaussian kernel in $\mathbb{R}^d$, and let $S_k(x^\star) \subseteq S$ be the set of k synthetic points in S nearest to $x^\star$. The Gen-LRA score is then

$$\hat{\mathcal{I}}(x^\star; R, S) \;=\; \sum_{s \in S_k(x^\star)} \left[\log \hat{p}_{R\cup\{x^\star\}}(s) - \log \hat{p}_R(s)\right]. \tag{2.6}$$

Our first theoretical result characterizes this quantity in closed form.

Theorem 2 (Closed-form approximation)**.** *Suppose $\hat{p}_R(s) \geq c > 0$ for all $s \in S_k(x^\star)$. Then, for any fixed k or any $k = o(m)$, the Gen-LRA score satisfies*

$$\hat{\mathcal{I}}(x^\star; R, S) \;=\; \frac{1}{m+1} \sum_{s \in S_k(x^\star)} \left[\frac{K_h(s - x^\star)}{\hat{p}_R(s)} - 1\right] \;+\; O_p(m^{-2}). \tag{2.7}$$

The proof (Appendix 2.A.2) follows from an exact identity for the augmented KDE together with a Taylor expansion of the logarithm. Theorem 2 shows that Gen-LRA is, to leading order, a sum of localized density-ratio-like quantities. The dominant term $K_h(s - x^\star)/\hat{p}_R(s)$ is large precisely when a synthetic point s is close to $x^\star$ (numerator) *and* lies in a region of low reference density (denominator). This is the fingerprint of memorization: training-induced synthetic points that cluster near $x^\star$ in regions the population would not naturally populate.

Comparison to DOMIAS DOMIAS [30] uses the scoring function $\hat{p}_S(x^\star)/\hat{p}_R(x^\star)$—a single density ratio evaluated at $x^\star$. Equation 2.7 shows that Gen-LRA evaluates an analogous ratio at k synthetic points correlated with memorization of $x^\star$, rather than at $x^\star$ itself. Gen-LRA thus aggregates evidence over a region where the memorization signal is expected to concentrate, while DOMIAS uses only a single-point estimate. This provides a formal explanation for why Gen-LRA and DOMIAS differ despite following the same threat model.

Role of localization The closed form in (2.7) makes clear why localization is essential. The summand is appreciable only at synthetic points s near $x^\star$; points far from $x^\star$ contribute $K_h(s - x^\star) \approx 0$ and thus only the -1 baseline term, which adds noise without signal. Restricting the sum to the k-nearest synthetic neighbors of $x^\star$ discards these noise-only terms and concentrates the statistic on the region where signal can exist. The parameter k thus serves as a bias-variance trade-off: larger k reduces variance by averaging more terms, while smaller k reduces bias by excluding neighbors outside the memorization region.

2.4.5 Power Under a Local Overfitting Model

Theorem 2 characterizes what Gen-LRA computes; it does not by itself guarantee that the computation separates members from non-members. We now establish that under a general model of local overfitting, the expected Gen-LRA score is provably larger for members than for non-members, and we characterize the conditions under which the gap is largest.

Definition 1 (Local overfitting)**.** A generator G trained on T exhibits *local overfitting at scale ρ with strength ϵ* if its induced density p_G decomposes as

$$p_G(s) \;=\; p(s) \;+\; \frac{\epsilon}{|T|} \sum_{x_i \in T} \phi_\rho(s; x_i) \;+\; r(s), \tag{2.8}$$

where p is the population distribution, $|T|$ denotes the cardinality of the training set, $\phi_\rho(\cdot\,; x_i)$ is a non-negative *excess mass function* concentrated within distance ρ of x_i with $\int \phi_\rho(s; x_i)\, ds = 1$, $\epsilon \in [0, 1]$ is the *overfitting strength*, and $r(s)$ is a residual satisfying $\int r(s)\, ds = -\epsilon$.

The formulation is deliberately general: ϕ_ρ can be any concentrated non-negative function, a sharp spike for exact replication, a diffuse bump for noisy memorization, or an asymmetric shape reflecting the generator's inductive bias, and r absorbs any displacement of mass from elsewhere in the distribution. The decomposition does not require a specific generator or model; it is purely a description of its output density.

Theorem 3 (Mean score gap)**.** *Let $\mu_1 = \mathbb{E}[\hat{\mathcal{I}}(x^\star; R, S) \mid x^\star \in T]$ and $\mu_0 = \mathbb{E}[\hat{\mathcal{I}}(x^\star; R, S) \mid x^\star \sim P]$ denote the expected Gen-LRA scores for members and non-members, respectively. Under Definition 1 and the regularity conditions of Theorem 2,*

$$\mu_1 - \mu_0 \;\gtrsim\; \frac{k\,\epsilon}{(m+1)\,|T|} \cdot \Psi(h, \rho, x^\star), \tag{2.9}$$

where the signal term *is*

$$\Psi(h, \rho, x^\star) \;=\; \mathbb{E}_s\left[\frac{K_h(s - x^\star)\,\phi_\rho(s; x^\star)}{\hat{p}_R(s)}\right], \tag{2.10}$$

and $\Psi(h, \rho, x^\star) > 0$ *whenever the support of* $\phi_\rho(\cdot; x^\star)$ *overlaps with the kernel* $K_h(\cdot - x^\star)$.

We defer the proof to Appendix 2.A.3. The argument proceeds by substituting the overfitting decomposition (2.8) into the leading-order expression from Theorem 2, and showing that the contributions from the population density p and the residual r cancel in the difference $\mu_1 - \mu_0$, leaving the self-overlap Ψ as the dominant signal.

Theorem 3 states that Gen-LRA has nontrivial power against any generator exhibiting local overfitting with $\epsilon > 0$. The size of the mean gap depends on three interpretable quantities, each governing a distinct aspect of the attack's behavior:

1. *Linear dependence on the overfitting strength* ϵ. The mean gap of Gen-LRA scales linearly with ϵ: generators that overfit more aggressively are more vulnerable to Gen-LRA, and generators that do not, for example those trained with strong differential privacy guarantees, to which ϕ_ρ must be nearly flat, should yield low-quality attacks.

2. *Localization to the memorization region.* The score in (2.7) approximates Ψ by summing the integrand over the k nearest synthetic neighbors of $x^\star$, so k controls how many synthetic points are included in the sum. When k is smaller than the number of synthetic points the generator places within distance $\sim \rho$ of $x^\star$, the sum ignores signal; when k is larger, the additional neighbors lie where $K_h(s - x^\star)$ is negligible and contribute only additional noise. The optimal k therefore depends on how the generator distributes its excess mass across synthetic points.

3. *Bandwidth-scale matching.* The signal term $\Psi(h, \rho, x^\star)$ is a kernel-overlap integral between K_h and the ϕ_ρ, and is non-negligible only in a neighborhood of $x^\star$ on the scale of ρ. The bandwidth h controls the resolution at which the KDE recovers the local density ratio $p_{R\cup\{x^\star\}}/p_R$ in this neighborhood: $h \gg \rho$ oversmooths and washes out the perturbation from adding $x^\star$, while $h \ll \rho$ produces an estimate of $\hat{p}_R$ that adds noise to the score. Because the attack signal is local, estimators with explicit control over a local smoothing scale are better attack instruments than estimators optimized for global density-estimation accuracy.

We study the empirical behavior of these quantities under controlled conditions in Section 2.5, and connect them to Gen-LRA's performance on real generators in Section 2.6.

2.4.6 Implementation Choices

The theoretical results above motivate each component of the Gen-LRA implementation (full pseudocode is given in Alg. 1).

Surrogate model We use Gaussian KDEs as the surrogate density estimator. This choice is motivated by Theorem 2: the closed-form expansion depends on Gaussian-kernel algebra, making Gen-LRA with KDEs analytically tractable and computationally cheap to run. KDEs are widely available and, per the remark following Theorem 3, their locally structured bias profile aligns with the task of detecting memorization at scale $\rho \approx h$. We include an ablation and discussion where we study replacing KDEs with a flow-based deep learning method (see Section 2.7.3), but find KDE empirically outperforms it. We set the bandwidth using Silverman's rule per results we find in Section 2.5. An ablation on encoding strategies for heterogeneous tabular data appears in Appendix 2.C.1; we that ordinal encoding for categorical variables outperforms other methods and thus choose it as a default.

Choice of k The choice of k governs a bias-variance trade-off (Section 2.4.4), and it plays different roles in the two settings Gen-LRA is used in. A realistic adversary cannot tune k against ground-truth membership labels and must commit to a single value; the ablation in Appendix 2.C.3 shows that $k = 10$ is robust across datasets and generators, and we adopt it as the default. A privacy auditor, by contrast, has access to the true membership labels by construction and seeks the tightest possible bound on the generator's vulnerability. For a given $x^\star$, the dominant cost is the $2k$ KDE evaluations against reference sets of size m and $m+1$, giving a per-query cost of $\mathcal{O}(k \cdot m \cdot d)$. Computing scores for all $k' \in \{1, \ldots, k\}$ requires no additional KDE evaluations: the per-neighbor log-density-ratio terms can be summed cumulatively in $\mathcal{O}(k)$ time. Sweeping k exhaustively up to $k_{\max}$ is therefore asymptotically equivalent to a single evaluation at $k_{\max}$, at cost $\mathcal{O}(k_{\max} \cdot m \cdot d)$. An auditor can thus report the strongest configuration over k, while our adversary defaults to $k = 10$.

Decision threshold The membership prediction rule (2.1) requires a decision threshold γ. In principle γ is tunable for a specific application, but $\hat{\mathcal{I}}(x^\star; R, S) > 0$ implies some degree of local overfitting to $x^\star$ and serves as a natural threshold.

Algorithm 1 Gen-LRA

Require: Test set X_{test}, synthetic set S, reference set R, locality k

Ensure: Membership scores $\{a_{x^\star}\}_{x^\star \in X_{\text{test}}}$

$\hat{p}_R \leftarrow \textsc{FitKDE}(R)$ ▷ Surrogate on reference only

for $x^\star \in X_{\text{test}}$ **do**

  $\hat{p}_{R\cup\{x^\star\}} \leftarrow \textsc{FitKDE}(R \cup \{x^\star\})$ ▷ Augmented surrogate

  $S_k(x^\star) \leftarrow \textsc{kNN}(S, x^\star, k)$ ▷ Localize to k-NN synthetic points

  $a_{x^\star} \leftarrow \sum_{s \in S_k(x^\star)} \left[\log \hat{p}_{R\cup\{x^\star\}}(s) - \log \hat{p}_R(s)\right]$

end for

return $\{a_{x^\star}\}_{x^\star \in X_{\text{test}}}$

2.5 Simulation Study

Before benchmarking Gen-LRA on real tabular generators (Section 2.6), we conduct a controlled simulation study to directly examine the structural properties of Theorems 2 and 3. The motivation is that real generators confound three questions: whether the attack works, under what conditions it works, and whether the parameters (ϵ, ρ) are meaningful in practice. A simulation lets us address all three under controlled conditions, and additionally study how the linearity, bandwidth-scale and localization properties from Section 2.4.5 manifest empirically.

2.5.1 Simulation Design

We directly instantiate the local overfitting model of Definition 1. Concretely, we fix a population distribution $P = \mathcal{N}(0, I_d)$ and sample T, a held-out non-member set, and R independently from P, each of size $n = 500$. To generate the synthetic set S, each of n synthetic points is drawn as

$$s_i \sim \begin{cases} P & \text{with probability } 1-\epsilon, \\ \mathcal{N}(x_j, \rho^2 I_d),\ x_j \sim \text{Unif}(T) & \text{with probability } \epsilon. \end{cases} \tag{2.11}$$

This directly implements Equation 2.8 with $\phi_\rho(\cdot; x_j) = \mathcal{N}(\cdot; x_j, \rho^2 I_d)$, making ϵ and ρ controllable configurations rather than latent properties of a trained model. We use $d = 40$, consistent with moderate-dimensional

tabular data.

Attacks. We evaluate Gen-LRA against two common threat-model-compatible baselines: DOMIAS [30], which computes a pointwise density ratio $\frac{\hat{p}_S(x^\star)}{\hat{p}_R(x^\star)}$, and DCR [10], which scores test points by distance to the nearest synthetic neighbor. We additionally include two variants of Gen-LRA, one with $k = 1$ (single-neighbor localization) and one with $k = 10$ (the default used in our benchmark experiments). Including both variants lets us isolate the role of the localization parameter k in shaping the attack's global and tail behavior.

Experimental grid. We compute $\epsilon \in \{0, 0.25, 0.5, 0.75, 1.0\}$ and $\rho \in \{0.1, 0.2, \ldots, 1.7\}$ over 10 random seeds per grid point. For each $(\epsilon, \rho, \text{seed})$ combination, we compute attack performance using Area Under the Curve (AUC) and True Positive Rate at the Fixed False Positive Rate of 0 (TPR@FPR=0) [11] by scoring all members and held-out non-members. This yields a measurement of both the global attack performance and the calibration of the classifier under various privacy leakage settings.

2.5.2 Simulation Results

Figure 2.2 presents four primary results. The top row reports mean AUC; the bottom row reports mean TPR@FPR=0 with respective error bars. The left column varies ϵ at fixed memorization scale; the right column varies ρ at fixed overfitting strength.

Linear Dependence on ϵ

Panels (a) and (c) report AUC and TPR@FPR=0 as functions of ϵ at fixed memorization scale $\rho = 1$. Both metrics scale linearly with ϵ for all four attacks, validating Theorem 3's prediction that the mean score gap is linear in the overfitting strength. Gen-LRA achieves the largest slope on both metrics, with $k = 10$ reaching AUC ≈ 0.81 at $\epsilon = 1$ on panel (a) and $k = 1$ and $k = 10$ both reaching TPR@FPR=0 ≈ 0.40 on panel (c).

The advantage over baselines is substantially larger for TPR@FPR =0 than AUC: at $\epsilon = 1$, Gen-LRA's TPR@FPR=0 is roughly 2× that of DCR and 4× that of DOMIAS, whereas the corresponding AUC ratios are smaller. This reflects the structural advantage of Gen-LRA's likelihood-ratio formulation. The theorem does not predict particular slope magnitudes, but the empirical ordering matches the structural argument in Section 2.4.4: aggregating evidence over k synthetic neighbors yields a stronger linear response than the single-point attacks DOMIAS and DCR.

Localization and Three Regimes of Memorization

Panels (b) and (d) sweep ρ at fixed $\epsilon = 0.5$. Recall that ρ is the scale of noise added to a training point when producing a memorized synthetic point: small ρ means synthetic points are nearly exact copies of training points, large ρ means they are heavily perturbed. The curves divide into three regimes that correspond to different types of empirical memorization.

Near-exact memorization ($\rho \lesssim 0.7$). Synthetic points memorized from training data are nearly identical to their source records. For a member $x^\star \in T$, this produces a single synthetic point s_1 that is a near-copy of $x^\star$, and adding $x^\star$ to the reference set creates a sharp, narrow peak in $\hat{p}_{R\cup\{x^\star\}}$ centered at $x^\star$. The likelihood ratio $\hat{p}_{R\cup\{x^\star\}}(s)/\hat{p}_R(s)$ carries useful signal only for synthetic points near x^*; elsewhere the two density estimates are nearly identical and the ratio is uninformative. DOMIAS, DCR, and Gen-LRA $k = 1$ target this peak. All three capture the full membership signal and achieve comparable AUC (≈ 0.69) and TPR@FPR=0 (≈ 0.40). Gen-LRA $k = 10$ underperforms on TPR@FPR=0 here as included additional synthetic neighbors are unrelated to $x^\star$ and their contributions effectively add noise to the score.

Approximate memorization ($\rho \in [0.7, 1.5]$). Synthetic points retain a recognizable relationship to training points but are spread out enough that no single synthetic point is a reliable copy of any given training point. The peak in $\hat{p}_{R\cup\{x^\star\}}$ is still localized to $x^\star$, but probability mass in S is no longer concentrated at x^*: instead, several synthetic points sit near $x^\star$, each carrying partial evidence. DOMIAS and DCR collapse in this regime. DOMIAS's single-point density ratio loses signal because the synthetic data is no longer concentrated at $x^\star$, and DCR loses signal because the closest synthetic point to a member is no longer reliably closer than the closest synthetic point to a non-member. Gen-LRA $k = 1$ also degrades but at a lower rate than DCR and DOMIAS. Gen-LRA $k = 10$ however, aggregates evidence across all the relevant synthetic neighbors and dominates on both AUC and TPR@FPR=0.

Diffuse memorization ($\rho \gtrsim 1.5$). Synthetic points are perturbed heavily. No local feature of S distinguishes regions near training points from regions near non-training points, and all four attacks collapse to random performance.

The Role of Bandwidth

Figure 2.3 repeats the ρ sweep from panels (b) and (d) for Gen-LRA at three bandwidths: 0.1×, 1×, and 10× Silverman's rule. The narrow (0.1×) and default (1×) bandwidths produce nearly indistinguishable curves on both AUC and TPR@FPR=0, indicating that Gen-LRA is robust to under-smoothing in this regime. The wide bandwidth (10×) collapses Gen-LRA's performance to near-random AUC and near-zero TPR@FPR=0 across all ρ. This corresponds with item (3) of Section 2.4.5: when $h \gg \rho$, the KDE oversmooths the

Table 2.1: Attack ranking performance across evaluation metrics over all 1,525 runs. Higher Top-k values are better, while lower mean rank indicates stronger performance. Gen-LRA achieves the highest Top-1 rate on every metric at more than 2× that of any baseline.

			TPR at Fixed FPR			
Method	Statistic	AUC	0.0	0.001	0.01	0.1
Gen LRA $k = 10$	Top-1	**38.6%**	**42.1%**	**41.5%**	**39.8%**	**46.2%**
	Top-3	**71.7%**	**78.5%**	**75.6%**	**73.1%**	**76.7%**
	Mean Rank (std)	**2.83 (2.12)**	**2.79 (2.05)**	**2.76 (2.02)**	**2.78 (2.04)**	**2.54 (2.0)**
Classifier	Top-1	16.0%	9.6%	9.1%	11.6%	9.6%
	Top-3	45.7%	35.4%	31.9%	35.2%	35.7%
	Mean Rank (std)	4.29 (2.57)	5.22 (2.16)	5.04 (2.36)	4.79 (2.47)	4.83 (2.48)
DCR	Top-1	9.0%	17.4%	15.0%	12.2%	8.3%
	Top-3	31.2%	44.1%	37.4%	37.9%	32.2%
	Mean Rank (std)	4.93 (2.34)	4.67 (2.3)	4.77 (2.44)	4.63 (2.39)	4.86 (2.3)
DCR-Diff	Top-1	12.0%	18.4%	16.4%	16.3%	16.3%
	Top-3	43.5%	60.3%	51.8%	50.0%	46.2%
	Mean Rank (std)	4.11 (2.21)	3.78 (2.04)	4.05 (2.35)	4.2 (2.49)	4.2 (2.48)
DOMIAS	Top-1	10.0%	12.7%	11.8%	11.3%	11.8%
	Top-3	36.8%	52.1%	43.4%	36.8%	41.0%
	Mean Rank (std)	4.42 (2.19)	4.22 (2.08)	4.56 (2.45)	5.03 (2.67)	4.63 (2.59)
DPI	Top-1	10.7%	2.4%	1.9%	4.0%	6.5%
	Top-3	38.9%	13.0%	12.4%	20.6%	29.9%
	Mean Rank (std)	4.5 (2.34)	6.79 (1.51)	5.94 (1.95)	5.18 (1.93)	4.97 (2.22)
LOGAN	Top-1	12.7%	16.2%	13.8%	11.3%	11.0%
	Top-3	28.7%	48.9%	41.7%	30.8%	32.1%
	Mean Rank (std)	5.51 (2.79)	4.35 (2.2)	4.71 (2.53)	5.3 (2.59)	5.19 (2.62)
Local Neighborhood	Top-1	2.4%	6.6%	5.8%	6.7%	4.4%
	Top-3	16.0%	32.9%	30.3%	29.9%	20.8%
	Mean Rank (std)	5.79 (1.99)	5.4 (2.09)	5.26 (2.28)	5.11 (2.3)	5.41 (2.14)
MC	Top-1	4.2%	12.5%	11.3%	8.5%	6.3%
	Top-3	22.5%	43.8%	35.7%	30.3%	25.3%
	Mean Rank (std)	5.61 (2.3)	4.71 (2.14)	4.91 (2.41)	5.11 (2.41)	5.32 (2.35)

perturbation introduced by adding $x^\star$ to R, washing out the local density ratio that Gen-LRA relies on; when $h \ll \rho$, the estimate is noisier but empirically retains the structural signal that the score depends on.

2.6 Experiments

The simulation study in Section 2.5 shows that different attacks and hyperparameter settings perform differently across memorization regimes. This raises a methodological concern: a small benchmark risks producing results that are artifacts of which regimes happen to dominate the selection, biasing the comparison toward attacks suited to those particular conditions. We therefore construct a large and diverse benchmark to assess Gen-LRA's effectiveness across a broad distribution of synthetic datasets: 35 tabular datasets sampled from OpenML [49], 9 tabular generative architectures, 9 membership inference attacks, and 5 random seeds. This yields 1,525 unique synthetic datasets and over 10,000 unique attack runs, spanning a wide range of memorization conditions that approximate what practitioners could encounter in deployment.

2.6.1 Benchmark Setup

For each dataset, following standard tabular synthetic data evaluation practice [50], we randomly partition the data without replacement into a training set T, holdout set H, and reference set R in an 80/10/10 ratio.

The generator is trained on T and produces a synthetic dataset S of size $|T|$. MIAs are then evaluated on a balanced test set composed of $\min(1000, |H|)$ members sampled from T and an equal number of non-members sampled from H. Full details on datasets, MIAs, and generators are provided in Appendix 2.B.

Gen-LRA and DOMIAS rely on density estimation, which we implement using Gaussian Kernel Density Estimation (KDE) with Silverman's rule for bandwidth selection. We find KDE outperforms deep learning based density estimators on this task for Gen-LRA (Section 2.7.3). Because KDE handles one hot encoded categorical data poorly, we use ordinal encoding for these attacks; an ablation across Principle Component Analysis and Variational Autoencoder based encodings appears in Appendix 2.C.1 and shows that ordinal encoding performs best. For all other attacks, numeric features are scaled and categorical features one hot encoded.

2.6.2 Baselines

We compare Gen-LRA with a fixed $k = 10$ against eight competing MIAs that operate under compatible threat models: LOGAN, MC, DCR, DCR-Diff, Classifier, Local Neighborhood, DOMIAS, and DPI [44, 45, 10, 15, 30, 13].

We evaluate across nine tabular generators spanning the major architectural families: PATEGAN ($\epsilon = 1$) for differentially private generation; Ads-GAN, CTGAN, and TVAE for adversarial training; Normalizing Flows for likelihood-based generation; ARF for tree-based generation; Tab-DDPM and TabSyn for diffusion-based generation; and RealTabFormer (RTF) for transformer-based generation [51, 1, 2, 38, 52, 53, 3, 54, 29]. For RealTabFormer and TabSyn we use the original implementations with default hyperparameters; for all other architectures we use the default Synthcity [50] implementations and settings.

All experiments were conducted on an AWS G5.2xlarge EC2 instance. The main benchmark required approximately 500 hours of compute, with an additional 80 hours used for the deep learning density estimation experiments described in Section 2.7.3.

2.7 Experiment Results

We summarize benchmark performance along two complementary axes: aggregate ranking across all 1,525 synthetic datasets, and performance restricted to the top 100 runs by each metric. The aggregate ranking measures how often each attack is the strongest available choice across diverse synthetic datasets a practitioner may encounter. The top-100 view measures attack performance in the synthetic datasets with the greatest amount of detected leakage.

Table 2.2: Mean performance with standard deviations across the top 100 runs for each metric. Bold indicates the highest value in each column. Despite outliers that inflate attacks that target near-memorization, Gen-LRA achieves the highest mean on every metric.

		TPR at Fixed FPR		
Method	AUC	0.0	0.01	0.1
Gen-LRA	**0.589 (0.03)**	**0.051 (0.03)**	**0.053 (0.03)**	**0.211 (0.05)**
Classifier	0.557 (0.05)	0.012 (0.02)	0.021 (0.02)	0.144 (0.06)
DCR	0.579 (0.04)	0.029 (0.04)	0.040 (0.04)	0.175 (0.06)
DCR-Diff	0.570 (0.04)	0.037 (0.04)	0.044 (0.04)	0.171 (0.05)
DOMIAS	0.546 (0.04)	0.024 (0.02)	0.024 (0.02)	0.140 (0.04)
DPI	0.526 (0.03)	0.001 (0.00)	0.013 (0.01)	0.114 (0.02)
LOGAN	0.493 (0.02)	0.013 (0.01)	0.015 (0.01)	0.105 (0.03)
Local N.	0.538 (0.04)	0.013 (0.02)	0.024 (0.02)	0.132 (0.04)
MC	0.548 (0.04)	0.016 (0.02)	0.024 (0.02)	0.134 (0.05)

2.7.1 Aggregate Ranking

An adversary attacking an unknown generator must commit to a single attack as they have no ground truth labels, so a relevant question is which attack most often delivers the strongest signal across a diverse sample of synthetic datasets. Table 2.1 reports, for each attack and metric, the fraction of runs in which the attack ranks first (Top-1), the fraction of runs in which it ranks in the top three (Top-3), and the mean rank across all 1,525 runs. The Top-1 rate answers this question directly, and Gen-LRA is the dominant choice. Gen-LRA achieves the highest Top-1 rate on every metric (38.6% on AUC, 42.1% on TPR@FPR=0, and 46.2% on TPR@FPR=0.1), more than 2× the rate of any baseline.

The closest competitors are DCR-Diff at 18.4% on TPR@FPR=0 and Classifier at 16.0% on AUC; no baseline exceeds a 19% Top-1 rate on any metric. Gen-LRA's margin widens at higher false-positive rates, reaching 2.8× the next-best attack at TPR@FPR=0.1, indicating that the benefit of aggregating evidence over k synthetic neighbors is most pronounced at operating points permitting a moderate false-positive budget. The supporting Top-3 rates (71.7%–78.5% for Gen-LRA, against $\leq$ 60.3% for any baseline) and mean ranks (2.54–2.83 for Gen-LRA, against $\geq$ 3.78 for any baseline) confirm that this dominance is consistent.

2.7.2 Performance on the Most Vulnerable Runs

Aggregate ranking measures an adversary's preference for an attack; we now turn to attack performance for runs that exhibited the highest detected leakage. Table 2.2 reports each metric's mean and standard deviation over the top 100 runs by that metric, the regime in which the adversary's attack choice has the largest practical consequence. Gen-LRA achieves the highest mean AUC (0.59) and the highest mean TPR

at every false-positive rate, with the largest gap at low FPR: Gen-LRA's mean TPR@FPR=0 of 0.05 is approximately 1.7× that of DCR-Diff and 2.5× that of DOMIAS.

We additionally report medians and interquartile ranges in the Appendix Table 2.6, which shows that the closest baseline performance is outlier driven. DCR and DCR-Diff have median TPR@FPR =0 values (0.011 and 0.024) well below their means, with IQRs touching zero. Gen-LRA's median TPR@FPR=0 of 0.055 exceeds its own mean, its IQR is bounded away from zero ($[0.034, 0.067]$), and its median AUC of 0.594 exceeds the third quartile of every baseline.

These outliers are concentrated on datasets generated by RealTabFormer, which accounts for 42 of the top 100 runs by AUC. Recently, [55] documented that RealTabFormer often produces near-exact copies of training records. The simulation in Section 2.5.2 identifies this small ρ regime as the one in which distance based baselines are most competitive with Gen-LRA at the fixed $k = 10$ used in our benchmark, implying that the top 100 runs overrepresent the regime least favorable to Gen-LRA, yet Gen-LRA leads on every metric in both mean and median.

2.7.3 Deep Learning Density Estimation

Gen-LRA estimates the likelihood of high dimensional data, a setting in which Gaussian Kernel Density Estimation (KDE) is typically outperformed by modern density estimators on metrics such as average negative log likelihood and negative evidence lower bound [56, 57]. It is therefore worth examining whether replacing KDE with a more expressive estimator improves attack performance. Following [30], we perform an ablation (details in Appendix 2.C.2) with Block Neural Autoregressive Flows (BNAF) [56] as the surrogate density estimator for Gen-LRA.

Figure 2.4 shows that KDE matches or outperforms BNAF on both AUC and TPR@FPR=0.01, with the largest gap on the generators that exhibit the most extreme privacy leakage. While BNAF produces a tighter distribution of scores across runs, this stability comes from a consistent failure to detect the strongest leakage signals: BNAF's upper tail tops out near 0.56 AUC, whereas KDE reaches above 0.65 on the same generators. The bandwidth analysis in Section 2.4.5 explains this result: the Gen-LRA signal is governed by the kernel overlap term $\Psi(h, \rho, x^\star)$, which requires the surrogate's local smoothing scale to match the memorization scale ρ. KDE controls this scale explicitly through its bandwidth. BNAF however optimizes for global density accuracy and exposes no analogous parameter, so its implicit smoothing scale is set by global training dynamics rather than matched to ρ. The result is a surrogate that is more expressive globally but less faithful in the local neighborhood where Gen-LRA's signal is found.

A second issue compounds the first: Gen-LRA compares two surrogates fit on R and $R \cup \{x^\star\}$, and

recovering the influence of a single point requires the difference between them to reflect that point rather than training stochasticity. KDE is deterministic given its bandwidth and data, so the two fits differ only in the contribution of $x^\star$. BNAF and other modern density estimators are trained by stochastic optimization, and the two fits differ in initialization, batch order, and optimizer state, which adds noise to the influence estimate of Gen-LRA. We therefore default to KDE for Gen-LRA, and recommend it to practitioners on the basis of empirical performance and substantially lower computational cost.

2.7.4 Privacy-Utility Trade-off

The preceding results compare attacks against each other; we now ask whether the privacy leakage detected by Gen-LRA tracks the quality of the synthetic data being audited. Figure 2.5 plots mean Gen-LRA AUC against three quality measures averaged across models for each benchmark run: XGBoost test AUC under a train-on-synthetic-test-on-real protocol (downstream utility), maximum mean discrepancy and Jensen-Shannon distance between real and synthetic samples (distributional fidelity). Lower MMD and JS values indicate higher fidelity, while higher XGBoost test AUC indicates higher utility.

PATE-GAN at $\epsilon = 1$ occupies the worst-quality corner of all three panels, with the lowest utility (XGBoost AUC ≈ 0.59) and the worst fidelity on both MMD and JS, and also the lowest Gen-LRA AUC (≈ 0.518). This is the expected behavior of differentially private synthesis: the privacy guarantee that limits Gen-LRA's signal also limits the synthetic data's resemblance to the population. At the opposite end, RealTabFormer, ARF, and TabSyn achieve the highest utility and the strongest fidelity scores while exhibiting the highest Gen-LRA AUC values in the benchmark. These generators produce synthetic data that supports downstream learning effectively, but the same fidelity that makes the synthetic data useful renders it more vulnerable to No-Box MIAs.

2.8 Discussion

2.8.1 Interpreting the Benchmark

Theorem 2 characterizes the Gen-LRA score as a sum of localized density ratios over the k synthetic neighbors of $x^\star$, and Theorem 3 shows that the expected score gap between members and non-members is governed by the kernel-overlap term $\Psi(h, \rho, x^\star)$ between the surrogate's bandwidth and the generator's memorization scale. The simulation in Section 2.5 instantiates this score in three regimes the theory makes precise. Under near-exact memorization, single-point statistics already capture the membership signal and aggregation over

k neighbors only adds noise. Under approximate memorization, signal is dispersed across multiple synthetic points and aggregation recovers what single-point statistics miss. Under diffuse memorization, no local statistic separates members from non-members, a regime in which all No-Box attacks must fail. Extending the model to capture non-uniform overfitting across training points, for instance, by allowing ϵ and ρ to vary with the local density of the population, is a natural refinement.

The distribution of overfitting behavior across modern tabular generators is not known a priori, and prior MIA evaluations have rarely been broad enough to characterize it. Most published tabular generative model MIA works span a handful of datasets and a small set of generator architectures, leaving open the question of whether reported performance generalizes beyond the conditions of the benchmark itself. Our evaluation spans 1,525 synthetic datasets across 9 generator families and 35 datasets, covering the architectural and statistical diversity a practitioner could realistically encounter.

Across this distribution, Gen-LRA achieves the highest Top-1 rate on every metric at more than 2× that of any baseline, indicating that the approximate-memorization regime predicted by Theorem 3 to favor Gen-LRA is well-represented in practice. RealTabFormer is the most prominent exception, with known near-exact memorization behavior that dominates the top-100 runs by AUC and accounts for the outlier-driven means observed for distance-based baselines. Even within this subset, Gen-LRA achieves the highest mean and median on every metric.

These numbers also understate Gen-LRA's true power. We fix $k = 10$ across all 1,525 runs rather than tuning it per dataset or per generator, but a privacy auditor has access to ground-truth membership labels by construction, and the cost analysis in Section 2.4.6 shows that sweeping k up to $k_{\max}$ is asymptotically equivalent to a single evaluation at $k_{\max}$. An auditor using Gen-LRA can therefore report the strongest configuration over k at no additional cost, yielding strictly tighter bounds than the fixed-k numbers reported here. The benchmark results should be read as a conservative estimate of what Gen-LRA delivers in practice.

The flip side is also informative: at fixed $k = 10$, Gen-LRA is *not* the top-ranked attack on roughly 60% of runs. No single attack, Gen-LRA included, is uniformly optimal across datasets and generator architectures in our benchmark. Whether a uniformly optimal No-Box MIA exists is an important open question with direct consequences for how synthetic data should be audited in practice.

2.8.2 No-Box Auditing Estimates Realistic Privacy Leakage

A common objection to No-Box MIAs is that they are weaker than shadow-box or query-based alternatives, and therefore underestimate privacy risk. This framing conflates two distinct questions. Privacy leakage is not a property of a model alone — it is a property of a model conditioned on a threat model. The relevant

question for a practitioner releasing synthetic data is not "what is the maximum leakage achievable by any conceivable adversary?" but "what leakage is achievable by an adversary I should plausibly defend against?"

Shadow and white-box attacks assume the adversary knows the generator's architecture or has direct access. A defender following standard release practice, withholding model details and releasing only the synthetic dataset, eliminates the information these attacks depend on. The leakage they measure is therefore not the leakage the release exposes, but the leakage that would be exposed under a counterfactual disclosure regime the defender did not adopt. The No-Box setting, by contrast, corresponds to the release itself: the adversary sees only the synthetic dataset and auxiliary population data they could plausibly obtain. Less conservative threat models remain essential for other auditing problems — verifying differential privacy guarantees, for instance, requires the strongest attacks available regardless of deployment realism [58]. But sample-level auditing under realistic adversaries is equally important, and progress in the No-Box regime, both theoretical and empirical, is what gives practitioners the tools to assess the privacy of the artifacts they actually release.

2.8.3 Limitations

Gen-LRA's theoretical guarantees and empirical performance come with three limitations, each suggesting a direction for future work.

Scope of the local overfitting model. Definition 1 characterizes the generator's output density as the population density plus additive excess mass concentrated within distance ρ of each training point, and Theorem 3 provides guarantees only for generators whose output density admits this decomposition with $\epsilon > 0$ and ρ comparable to the surrogate's bandwidth. The model further idealizes ϕ_ρ and ϵ as uniform across training points, whereas in practice memorization is non-uniform: rare records, outliers, and points in low-density regions are typically memorized more strongly than the rest. The model is nonetheless rich enough to capture the three regimes that distinguish Gen-LRA's behavior from single-point baselines (Section 2.5), and its predictions are consistent with Gen-LRA's empirical performance on real generators.

Per-target hyperparameter sub-optimality. The non-uniformity of memorization carries through to Gen-LRA itself: a single choice of bandwidth h and locality k is unlikely to be simultaneously optimal for every $x^\star$, and per-target attack power varies with the local memorization geometry around each test point. Our work does not disentangle this per-target variation. That said, the cost analysis in Section 2.4.6 establishes that adaptive per-target configurations are computationally feasible, making this a natural and tractable direction for future work rather than an obstacle.

Surrogate scaling. KDE is known to scale poorly with dimensionality, and the empirical observation that KDE outperforms BNAF in our benchmark relies on the moderate dimensions ($d \lesssim 100$ after encoding) of the OpenML datasets we evaluate. On substantially higher-dimensional tabular data, neither KDE nor BNAF may be adequate, and Gen-LRA's surrogate component would require revisiting. The density-ratio-estimation literature [59, 60] establishes that estimating a ratio directly is more sample-efficient than estimating two densities and dividing them, suggesting an attractive alternative to the two surrogate fits in Algorithm 1. Adapting these methods to Gen-LRA is non-trivial: standard direct-ratio estimators assume two independent samples drawn from distinct distributions, whereas the two distributions whose ratio we require are induced by reference sets that differ by a single point. Closing this gap likely requires new estimators tailored to the influence-function structure of the problem, which we view as a promising direction for scaling Gen-LRA to higher-dimensional regimes.

2.9 Conclusion

Our results suggest that the dominant No-Box membership inference attacks for synthetic data release underestimate the privacy risk of the artifacts they evaluate. DOMIAS tests *the wrong hypothesis*, evaluating where synthetic density exceeds reference density rather than whether a specific point influenced the synthetic distribution; and distance-based methods such as DCR detect leakage only when memorization is near-exact. Both classes underestimate in the approximate-memorization regime, which our benchmark shows is where most modern tabular generators likely operate. A practitioner concluding that a release is private on the basis of these attacks is relying on measurements whose inductive biases are matched to a specific overfitting phenomena that the generator may not exhibit.

Gen-LRA addresses this gap with an attack that is both principled and practical. The likelihood-ratio influence-function formulation directly tests the membership hypothesis, the closed-form characterization (Theorem 2) makes explicit what the attack measures, and the mean-score-gap result (Theorem 3) provides testable predictions about when it succeeds, predictions we validate in a controlled simulation and that align with empirical performance across 1,525 synthetic datasets spanning nine generators. Gen-LRA achieves the highest Top-1 rate on every metric at more than 2× that of any baseline, and corresponding highest mean values in the top 100 highest privacy leakage runs.

The framework we develop suggests several directions for future work. Adaptive bandwidth and locality selection on a per-target basis would likely increase attack performance, particularly for the long-tail records that are memorized most strongly. Replacing the two separate density estimates in Algorithm 1 with a direct density-ratio estimator tailored to the single-point-difference setting offers a route to scaling Gen-LRA beyond

the moderate-dimensional regimes where KDE remains effective, and would likely improve performance on our benchmark as well. Extending the local-overfitting model and the influence-function framing to settings where synthetic tabular data release is becoming common such as longitudinal records is a natural next step. Finally, Gen-LRA's broad but non-uniform dominance leaves open whether a uniformly optimal No-Box MIA exists—a question we view as central to the privacy auditing literature, since such an attack would let practitioners directly certify a synthetic dataset's privacy under the threat model rather than relying on guarantees from defensive frameworks.

2.10 Ethical Considerations

This work proposes a membership inference attack against synthetic tabular data, which is itself a re-identification method. We address the resulting ethical considerations along three dimensions: the risks the attack poses, the benefits it provides, and the steps we have taken to ensure that the latter outweigh the former.

Risks. Adversaries who can infer whether an individual's record was used to train a generative model pose direct privacy risks to that individual, particularly in domains such as healthcare, finance, and the social sciences where sensitive personal data is routinely used. A synthetic dataset that fails to obfuscate membership information can enable re-identification of training-set participants, undermining the privacy guarantees that motivate synthetic data release in the first place. Gen-LRA, by improving the state of the art in No-Box membership inference, raises the ceiling on what such an adversary can achieve.

Benefits. The same capability that creates risk also enables stronger privacy auditing. Practitioners releasing synthetic datasets currently rely on similarity-based heuristics that have been shown to be inadequate proxies for privacy, or on attacks whose threat-model assumptions do not match realistic release scenarios. Gen-LRA gives auditors a principled, computationally tractable tool to assess sample-level privacy leakage under the threat model that actually corresponds to public release. Without attacks of this kind, defenders cannot measure the privacy of the artifacts they release.

Steps taken to minimize harm. We evaluate Gen-LRA exclusively on publicly available datasets from OpenML, which contain no personally identifying information beyond what their original releasers have already chosen to publish. We do not target any deployed synthetic data release, nor do we attempt to re-identify individuals in any real-world dataset. The generative models we attack are trained by us on these public datasets specifically for the purpose of evaluating Gen-LRA, so no third party's privacy claims are challenged by our experiments. We release our code openly to support reproducibility and to enable defenders to audit their own releases, which we view as the primary use case. Responsible disclosure does not apply

here, as no specific deployed system is implicated; the vulnerability we characterize is structural to overfitting in tabular generative models and has been documented in prior work.

We believe adversarial work of this kind is essential to the development of trustworthy synthetic data release. Deferring the publication of stronger attacks would leave practitioners auditing with tools that systematically underestimate privacy risk, which we view as a net riskier outcome.

2.A Proofs and Supporting Results

This appendix provides proofs for the three theoretical results of Section 2.4.

2.A.1 Proof of Theorem 1 (Invariance)

Proof. Let $g : \mathcal{X} \to \mathcal{X}$ be a continuously differentiable invertible function with Jacobian $J(x) = \frac{dg}{dx}(x)$, where $|J(x)| \neq 0$ for all $x \in \mathcal{X}$. For sets $A \subseteq \mathcal{X}$, write $\tilde{A} = g(A)$, and for a density p let $\tilde{p}$ denote the density induced on the transformed space.

By the change-of-variables formula for probability densities,

$$\tilde{p}(g(A)) \;=\; \frac{p(A)}{|J(A)|}, \tag{2.12}$$

where $|J(A)|$ denotes the product of Jacobian determinants evaluated on A (treating J as acting pointwise on a set of samples).

Applying this to the conditional density in the numerator of the log-likelihood ratio:

$$\begin{aligned} \tilde{p}(\tilde{S} \mid \tilde{R} \cup \{\tilde{x}^\star\}) \;&=\; \frac{\tilde{p}(\tilde{S}, \tilde{R} \cup \{\tilde{x}^\star\})}{\tilde{p}(\tilde{R} \cup \{\tilde{x}^\star\})} && (2.13) \\ &=\; \frac{p(S, R \cup \{x^\star\})/|J(S, R \cup \{x^\star\})|}{p(R \cup \{x^\star\})/|J(R \cup \{x^\star\})|}. && (2.14) \end{aligned}$$

Similarly for the denominator:

$$\tilde{p}(\tilde{S} \mid \tilde{R}) \;=\; \frac{\tilde{p}(\tilde{S}, \tilde{R})}{\tilde{p}(\tilde{R})} \;=\; \frac{p(S, R)/|J(S, R)|}{p(R)/|J(R)|}. \tag{2.15}$$

Taking the ratio of these two conditional densities:

$$\frac{\tilde{p}(\tilde{S} \mid \tilde{R} \cup \{\tilde{x}^\star\})}{\tilde{p}(\tilde{S} \mid \tilde{R})} = \frac{p(S, R \cup \{x^\star\})}{p(R \cup \{x^\star\})} \cdot \frac{p(R)}{p(S, R)} \cdot \underbrace{\frac{|J(R \cup \{x^\star\})| \cdot |J(S, R)|}{|J(S, R \cup \{x^\star\})| \cdot |J(R)|}}_{=1}. \tag{2.16}$$

The Jacobian factors cancel because the joint samples $(S, R \cup \{x^\star\})$ and (S, R) differ by a single point whose Jacobian appears identically in the numerator of one ratio and the denominator of the other. This leaves

$$\frac{\tilde{p}(\tilde{S} \mid \tilde{R} \cup \{\tilde{x}^\star\})}{\tilde{p}(\tilde{S} \mid \tilde{R})} = \frac{p(S \mid R \cup \{x^\star\})}{p(S \mid R)}. \tag{2.17}$$

Taking logarithms yields $\mathcal{I}(g(x^\star), g(R), g(S)) = \mathcal{I}(x^\star, R, S)$. □

2.A.2 Proof of Theorem 2 (Closed-Form Approximation)

Proof. Let $\hat{p}_R$ denote a Gaussian KDE fit on R with bandwidth h, where $|R| = m$. The proof proceeds in four steps.

Step 1: Exact decomposition of the augmented KDE. By the definition of the Gaussian KDE,

$$\begin{aligned} \hat{p}_{R \cup \{x^\star\}}(s) &= \frac{1}{m+1} \sum_{x \in R \cup \{x^\star\}} K_h(s - x) \\ &= \frac{m}{m+1} \hat{p}_R(s) + \frac{1}{m+1} K_h(s - x^\star). \end{aligned} \tag{2.18}$$

This identity is exact; no approximation is used.

Step 2: Ratio form. Dividing both sides by $\hat{p}_R(s)$,

$$\begin{aligned} \frac{\hat{p}_{R \cup \{x^\star\}}(s)}{\hat{p}_R(s)} &= \frac{m}{m+1} + \frac{1}{m+1} \cdot \frac{K_h(s - x^\star)}{\hat{p}_R(s)} \\ &= 1 + \frac{1}{m+1} \left[\frac{K_h(s - x^\star)}{\hat{p}_R(s)} - 1 \right]. \end{aligned} \tag{2.19}$$

Step 3: Taylor expansion of the logarithm. Let

$$u_s := \frac{1}{m+1} \left[\frac{K_h(s - x^\star)}{\hat{p}_R(s)} - 1 \right]. \tag{2.20}$$

The regularity assumption $\hat{p}_R(s) \geq c > 0$ on $S_k(x^\star)$, together with the boundedness of the Gaussian kernel K_h, implies that $|u_s| \leq C/(m+1)$ for a constant C depending on c, h, and d. For $|u| \to 0$, the Taylor

expansion

$$\log(1+u) \;=\; u - \tfrac{u^2}{2} + O(u^3) \tag{2.21}$$

gives

$$\log \frac{\hat{p}_{R\cup\{x^\star\}}(s)}{\hat{p}_R(s)} \;=\; \frac{1}{m+1}\left[\frac{K_h(s-x^\star)}{\hat{p}_R(s)} - 1\right] + O(m^{-2}). \tag{2.22}$$

Step 4: Summation over the local neighborhood. Summing over $s \in S_k(x^\star)$:

$$\begin{aligned}\hat{\mathcal{I}}(x^\star; R, S) \;&=\; \sum_{s\in S_k(x^\star)} \log \frac{\hat{p}_{R\cup\{x^\star\}}(s)}{\hat{p}_R(s)} \\ &=\; \frac{1}{m+1} \sum_{s\in S_k(x^\star)} \left[\frac{K_h(s-x^\star)}{\hat{p}_R(s)} - 1\right] + O(k\, m^{-2})\end{aligned} \tag{2.23}$$

For fixed k or $k = o(m)$, the remainder is $O_p(m^{-2})$, which completes the proof. □

2.A.3 Proof of Theorem 3 (Mean Score Gap)

Proof sketch. We work throughout with the leading-order expression from Theorem 2:

$$\hat{\mathcal{I}}(x^\star; R, S) \;\approx\; \frac{1}{m+1} \sum_{s\in S_k(x^\star)} \left[\frac{K_h(s-x^\star)}{\hat{p}_R(s)} - 1\right]. \tag{2.24}$$

The proof proceeds in four steps.

Step 1: Expected score. Let $g(s, x^\star) := K_h(s-x^\star)/\hat{p}_R(s) - 1$. Taking expectations over the randomness in S:

$$\mathbb{E}[\hat{\mathcal{I}}(x^\star; R, S)] \;\approx\; \frac{k}{m+1} \cdot \mathbb{E}_{s\sim p_G}\left[g(s, x^\star) \,\middle|\, s \in S_k(x^\star)\right]. \tag{2.25}$$

The k-NN localization concentrates s near $x^\star$, where the kernel $K_h(s-x^\star)$ is appreciable; outside this region $g(s, x^\star) \approx -1$ and contributes only to the constant baseline. For the purpose of comparing μ_1 and μ_0, we can treat the expectation as being taken with respect to p_G itself; localization affects μ_0 and μ_1 symmetrically and does not contribute to their difference.

Step 2: Substitution of the overfitting decomposition. Plugging Equation 2.8 into $\mathbb{E}_{s\sim p_G}[K_h(s-$

$x^\star)/\hat{p}_R(s)]$ yields three additive pieces:

$$\mathbb{E}_{s\sim p_G}\left[\frac{K_h(s-x^\star)}{\hat{p}_R(s)}\right] = \underbrace{\int \frac{K_h(s-x^\star)}{\hat{p}_R(s)}\, p(s)\, ds}_{\text{Piece A: population}} + \underbrace{\frac{\epsilon}{|T|}\sum_{i=1}^{|T|}\int \frac{K_h(s-x^\star)}{\hat{p}_R(s)}\, \phi_\rho(s;x_i)\, ds}_{\text{Piece B: memorization sum}} + \underbrace{\int \frac{K_h(s-x^\star)}{\hat{p}_R(s)}\, r(s)\, ds}_{\text{Piece C: residual}}. \tag{2.26}$$

Step 3: Differencing member vs. non-member expectations. We examine how each piece changes between the hypotheses H_0 ($x^\star \sim P$, independent of T) and H_1 ($x^\star \in T$).

Piece A is a functional of $x^\star$ and the population density p only; it does not reference T. Under both hypotheses, $x^\star$ has marginal distribution P, so Piece A contributes identically to μ_0 and μ_1 and cancels in the difference. The same argument applies to *Piece C*: the residual r is a property of the generator's output density, not of $x^\star$'s membership status.

For *Piece B*, define the kernel-overlap integral

$$f(x^\star, x_i) \;:=\; \int \frac{K_h(s-x^\star)}{\hat{p}_R(s)}\, \phi_\rho(s;x_i)\, ds, \tag{2.27}$$

which measures the overlap between the kernel centered at $x^\star$ and the memorization bump centered at x_i. This integral is largest when $x_i = x^\star$; we denote this *self-overlap* by $\Psi(h,\rho,x^\star) := f(x^\star,x^\star)$. Piece B then reads $\frac{\epsilon}{|T|}\sum_{i=1}^{|T|} f(x^\star,x_i)$, and we compare this sum under the two hypotheses.

Let $B(x^\star) := \sum_{i:x_i\neq x^\star} f(x^\star,x_i)$ denote the *background sum*: the total overlap between the kernel at $x^\star$ and memorization bumps centered at training points *other than* $x^\star$. Then:

- *Under* H_0 ($x^\star \sim P$, independent of T): none of the x_i equals $x^\star$, so $\sum_{i=1}^{|T|} f(x^\star,x_i) = B(x^\star)$, a sum of $|T|$ terms.
- *Under* H_1 ($x^\star = x_j$ for some j): the sum decomposes as

$$\sum_{i=1}^{|T|} f(x^\star,x_i) \;=\; \underbrace{f(x^\star,x_j)}_{=\Psi(h,\rho,x^\star)} \;+\; \underbrace{\sum_{i\neq j} f(x^\star,x_i)}_{=B(x^\star)}, \tag{2.28}$$

  where the first term is the self-overlap guaranteed by the memorization bump $\phi_\rho(\cdot;x_j)$ sitting on top of the kernel at $x^\star = x_j$, and the second term is a background sum of $|T|-1$ terms.

The two versions of $B(x^\star)$ differ by a single term out of $|T|$, which is $O(1/|T|)$ relative to the full background; to leading order they are equal. The dominant contribution to $\mu_1 - \mu_0$ is therefore the self-overlap term, which is present only under H_1:

$$\begin{aligned}\mu_1 - \mu_0 &\gtrsim \frac{k}{m+1} \cdot \frac{\epsilon}{|T|} \cdot \Psi(h, \rho, x^\star) \\ &= \frac{k\,\epsilon}{(m+1)\,|T|} \cdot \mathbb{E}_s\left[\frac{K_h(s - x^\star)\,\phi_\rho(s; x^\star)}{\hat{p}_R(s)}\right]. \end{aligned} \tag{2.29}$$

Step 4: Positivity of the signal term. The integrand of $\Psi(h, \rho, x^\star)$ is a product of non-negative quantities. It is strictly positive whenever the supports of $\phi_\rho(\cdot; x^\star)$ and $K_h(\cdot - x^\star)$ intersect—i.e., whenever the attacker's bandwidth reaches the memorization region. □

A member $x^\star \in T$ exhibits extra local synthetic density compared to a non-member because the memorization bump $\phi_\rho(\cdot; x^\star)$ is placed at $x^\star$'s own location. All other contributions to local synthetic density, the population term, the residual, and memorization bumps placed at other training points, occur symmetrically for members and non-members (both of which are distributed according to P) and cancel in the difference. The attack's power thus derives entirely from the self-overlap signal.

2.B Experiments/ Replication Details

2.B.1 MIAs for Generative Models Descriptions

The Membership Inference Attacks referenced in this paper is are described as follows:

Distance to Closest Record (DCR / DCR-Diff). A common class of membership inference attacks relies on the hypothesis that synthetic generators may reproduce training records or place generated points unusually close to them in feature space [10]. Under this intuition, query points corresponding to members should exhibit smaller distances to the synthetic dataset than non-members. The Distance to Closest Record (DCR) attack quantifies this effect using the score

$$f_{\mathrm{DCR}}(x^*, S) = -\min_{x \in S} d(x^*, x),$$

where $d(\cdot, \cdot)$ denotes a chosen distance function. A natural extension, DCR-Diff, incorporates a reference

dataset R to normalize this proximity signal by comparing closeness to both datasets:

$$f_{\text{DCR}}(x^*, S, R) = -\min_{x \in S} d(x^*, x) - \min_{x \in R} d(x^*, x).$$

DOMIAS / Density Estimation. DOMIAS [30] approaches membership inference from a distributional perspective by comparing how likely a query record is under the synthetic distribution relative to a reference distribution. Specifically, it evaluates the density ratio

$$f_{\text{DOMIAS}}(x^*, S, R) = \frac{p_S(x^*)}{p_R(x^*)},$$

where p_S and p_R are density estimates fit to the synthetic and reference datasets, respectively. In practice, these densities may be estimated using approaches such as kernel density estimation or neural density models. A simpler related baseline proposed in [15] uses only the synthetic density estimate,

$$f_{\text{Density Estimate}}(x^*, S) = p_S(x^*),$$

without explicitly contrasting against a reference distribution.

Data Plagiarism Index (DPI) / Local Neighborhood. The Data Plagiarism Index (DPI) [13] focuses on localized memorization by examining the composition of the neighborhood surrounding a query point. Given x^*, the method constructs a k-nearest-neighbor set $D(x^*)$ drawn jointly from synthetic and reference data, and computes the ratio of synthetic to reference neighbors:

$$f_{\text{DPI}}(x^*, S, R) = \frac{\sum_{\mathbf{z} \in D(x^*)} \mathbb{I}(\mathbf{z} \in S)}{\sum_{\mathbf{z} \in D(x^*)} \mathbb{I}(\mathbf{z} \in R)}.$$

Higher values indicate stronger local resemblance to the synthetic data distribution. A closely related local attack in [15] replaces the fixed-k neighborhood with all points falling within a specified radius of x^*.

LOGAN / Classifier-based. LOGAN was first introduced as a white-box membership inference attack [44] and later adapted to the black-box setting in [30]. The core idea is to train a classifier to distinguish samples from the target synthetic dataset S and a reference dataset R. In the original formulation, this classifier is implemented as the discriminator of a GAN trained on synthetic data, and the resulting score

$$f_{\text{LOGAN}}(x^*, S, R) = D_\theta(x^*)$$

is used as the membership signal. Records receiving higher discriminator confidence are considered more

likely to be members. Subsequent work [15] generalized this approach by replacing the GAN discriminator with standard supervised classifiers such as random forests.

2.B.2 Generative Model Architecture Descriptions

In all experiments, we use the implementations of these models from the Python package Synthcity [50]. For benchmarking purposes, we use the default hyperparameters for each model. A brief description of each model is as follows:

- **CTGAN** [38]: Conditional Tabular Generative Adversarial Network uses a GAN framework with conditional generator and discriminator to capture multi-modal distributions. It employs mode normalization to better learn mixed-type distributions.
- **TVAE** [38]: Tabular Variational Auto-Encoder is similar to CTGAN in its use of mode normalizing techniques, but instead of a GAN architecture, it employs a Variational Autoencoder.
- **Normalizing Flows (NFlows)** [52]: Normalizing flows transform a simple base distribution (e.g., Gaussian) into a more complex one matching the data by applying a sequence of invertible, differentiable mappings.
- **Bayesian Network (BN)** [51]: Bayesian Networks use a Directed Acyclic Graph to represent the joint probability distribution over variables as a product of marginal and conditional distributions. It then samples the empirical distributions estimated from the training dataset.
- **Adversarial Random Forests (ARF)** [53]: ARFs extend the random forest model by adding an adversarial stage. Random forests generate synthetic samples which are scored against the real data by a discriminator network. This score is used to re-train the forests iteratively.
- **Tab-DDPM** [3]: Tabular Denoising Diffusion Probabilistic Model adapts the DDPM framework for image synthesis. It iteratively refines random noise into synthetic data by learning the data distribution through gradients of a classifier on partially corrupted samples with Gaussian noise.
- **PATEGAN** [1]: The PATEGAN model uses a neural encoder to map discrete tabular data into a continuous latent representation which is sampled from during generation by the GAN discriminator and generator pair.
- **Ads-GAN** [2]: Ads-GAN uses a GAN architecture for tabular synthesis but also adds an identifiability metric to increase its ability to not mimic training data.

- **TabSyn** [54]: TabSyn uses a Variational Auto-Encoder to learn a latent space in which it builds a diffusion model from. TabSyn usually achieves state of the art data quality metrics relative to other methods compared.

2.B.3 Benchmarking Datasets References

We provide the URL for the sources of each dataset considered in the paper.

2.C Ablations

2.C.1 Gen-LRA Encoding

As our main experiment uses Kernel Density Estimation (KDE) over (usually) heterogeneous datasets, we present an ablation for encoding tabular data to be numeric such that KDE can converge. We experiment with 3 common strategies used in the density estimation literature: ordinal encoding for categorical variables, one-hot encoding categorical variables and then performing Principle Component Analysis (PCA), and using a Variational Auto-Encoder to learn continuous latent representations of the data.

We repeat our ablation subset experiment with these three encoding schemes. For PCA we use the number of eigenvectors that explain up to 95 %variance and for the VAE encoding we use TabSyn's original auto-encoder with default settings. Overall, we evaluate the top 100 runs for each metric and find that Ordinal encodings yield the best results (see Table 2.4).

2.C.2 Deep Learning Estimation

We replace the KDE surrogate in Gen-LRA with Block Neural Autoregressive Flows (BNAF) [56], following the implementation and hyperparameters of [30]. Each BNAF model is trained with default implementation parameters with early stopping on a held-out validation split of the reference set, using the Adam optimizer with default learning rate. We fit one BNAF on R and a second on $R \cup \{x^\star\}$ for each test point and compute the Gen-LRA score using these two surrogates in place of the corresponding KDEs. To induce greater overfitting in the trained generators and thereby produce stronger membership signal for both surrogates to recover, we reduce the training, holdout, reference, and synthetic set sizes to 250 records each. We then evaluate the top 100 highest-scored runs for each metric from the ordinal-encoded KDE and BNAF versions of Gen-LRA. All other aspects of the small-scale benchmark protocol, including datasets, generators, and attack baselines, match Section 2.C. Results and analysis appear in Section 2.7.3.

Table 2.3: List of Datasets included in Section 5.

Dataset	OpenML ID	N-size	Classes	Cat. Feat.	Num Feat.
GesturePhaseSegmentationProcessed	4538	9873	5	1	32
MiceProtein	40966	1080	8	5	77
PhishingWebsites	4534	11055	2	31	0
adult	1590	48842	2	9	6
analcatdata_authorship	40983	4839	2	1	5
bank-marketing	1461	45211	2	10	7
banknote-authentication	1462	1372	2	1	4
blood-transfusion-service-center	1464	748	2	1	4
car	40975	1728	4	7	0
churn	40701	5000	2	5	16
climate-model-simulation-crashes	1467	540	2	1	20
cmc	23	1473	3	8	2
connect-4	40668	67557	3	43	0
credit-approval	29	690	2	10	6
credit-g	31	1000	2	14	7
diabetes	37	768	2	1	8
electricity	151	45312	2	2	7
eucalyptus	43924	736	5	15	5
kc1	1067	2109	2	1	21
kc2	1063	522	2	1	21
kr-vs-kp	3	3196	2	37	0
letter	6	20000	26	1	16
mfeat-morphological	18	2000	10	1	6
numerai28.6	23517	96320	2	1	21
optdigits	28	5620	10	1	64
pc3	1044	10936	3	4	24
pendigits	32	10992	10	1	16
phoneme	1489	5404	2	1	5
satimage	182	6430	6	1	36
segment	40984	2310	7	1	19
sick	38	3772	2	23	7
spambase	44	4601	2	1	57
steel-plates-fault	40983	4839	2	1	5
texture	40499	5500	11	1	40
tic-tac-toe	50	958	2	10	0
vehicle	54	846	4	1	18

Table 2.4: Mean (std) LRA detection performance by encoding strategy, top 100 runs per metric, averaged over datasets, generative methods, and seeds. Rows sorted by AUC-ROC. Bold marks best encoding per metric.

Encoding	AUC-ROC	TPR@FPR=0	TPR@FPR=0.001	TPR@FPR=0.01	TPR@FPR=0.1
Ordinal	**0.616 (0.053)**	**0.011 (0.010)**	**0.013 (0.012)**	**0.043 (0.028)**	**0.221 (0.079)**
VAE	0.594 (0.049)	0.009 (0.010)	0.012 (0.012)	0.039 (0.027)	0.198 (0.064)
PCA	0.575 (0.046)	0.010 (0.008)	0.012 (0.009)	0.031 (0.021)	0.171 (0.056)

Table 2.5: Mean AUC-ROC at different k values for Gen-LRA for top 300 highest runs.

Model	$k = 1$	$k = 5$	$k = 10$	$k = 25$	$k = 50$	$k = 100$
AdsGAN	0.57 (0.00)	0.58	0.57 (0.00)	0.57 (0.01)	0.58 (0.01)	0.61 (0.00)
ARF	0.60 (0.02)	0.58 (0.02)	0.59 (0.01)	0.58 (0.01)	0.58 (0.01)	0.58 (0.01)
CTGAN	0.60 (0.01)	0.58 (0.00)	0.59 (0.02)	0.60 (0.00)	0.58 (0.01)	0.58 (0.01)
Tab-DDPM	0.60 (0.02)	0.60 (0.01)	0.60 (0.02)	0.61 (0.00)	0.60 (0.01)	0.60 (0.01)
N-Flow	0.61 (0.00)	0.57 (0.01)	0.59 (0.00)	0.59 (0.00)	0.59 (0.00)	0.60 (0.01)
RTF	0.59 (0.02)	0.59 (0.02)	0.59 (0.02)	0.60 (0.02)	0.59 (0.02)	0.59 (0.02)
TabSyn	0.62 (0.03)	0.61 (0.02)	0.60 (0.02)	0.60 (0.01)	0.59 (0.01)	0.58 (0.01)
TVAE	0.60 (0.00)	0.59 (0.00)	0.57 (0.00)	0.57 (0.00)	0.58 (0.01)	0.58 (0.01)

Table 2.6: Median with interquartile ranges for MIAs across the top 100 runs for each metric. Distance-based baselines (DCR, DCR-Diff) exhibit medians well below their means, with IQRs touching zero, indicating that their strong-run performance is driven by a small set of near-exact memorization outliers. Gen-LRA's median exceeds its own mean and its IQR is bounded away from zero, with median AUC (0.594) exceeding the third quartile of almost every baseline.

		TPR at Fixed FPR			
Method	AUC	0.0	0.001	0.01	0.1
Gen-LRA	**0.594 [0.574, 0.609]**	**0.055 [0.034, 0.067]**	**0.055 [0.034, 0.067]**	**0.055 [0.038, 0.072]**	**0.214 [0.175, 0.237]**
Classifier	0.559 [0.524, 0.597]	0.005 [0.000, 0.017]	0.005 [0.001, 0.018]	0.015 [0.007, 0.027]	0.133 [0.105, 0.181]
DCR	0.579 [0.561, 0.600]	0.011 [0.000, 0.053]	0.012 [0.002, 0.053]	0.022 [0.011, 0.063]	0.178 [0.125, 0.212]
DCR-Diff	0.576 [0.541, 0.591]	0.024 [0.008, 0.063]	0.025 [0.008, 0.063]	0.031 [0.013, 0.072]	0.169 [0.131, 0.205]
DOMIAS	0.546 [0.523, 0.568]	0.019 [0.009, 0.034]	0.019 [0.009, 0.035]	0.020 [0.008, 0.032]	0.132 [0.110, 0.172]
DPI	0.528 [0.511, 0.543]	0.000 [0.000, 0.000]	0.001 [0.001, 0.001]	0.011 [0.010, 0.013]	0.113 [0.101, 0.123]
LOGAN	0.495 [0.474, 0.507]	0.007 [0.002, 0.020]	0.006 [0.002, 0.020]	0.010 [0.002, 0.026]	0.107 [0.082, 0.123]
Local Neighborhood	0.534 [0.505, 0.557]	0.005 [0.000, 0.012]	0.006 [0.001, 0.015]	0.014 [0.010, 0.029]	0.118 [0.102, 0.145]
MC	0.553 [0.522, 0.571]	0.006 [0.000, 0.023]	0.006 [0.002, 0.024]	0.017 [0.009, 0.031]	0.121 [0.100, 0.165]

2.C.3 Ablation: Different k sizes

Gen-LRA targets local overfitting by utilizing the k-nearest neighbors in S to x^*. Consequently, k serves as a hyperparameter in the attack. To assess the impact of k on attack efficacy, we replicate the benchmarking experiments from Section 2.6 across varying values of k. The average AUC-ROC and corresponding standard deviations for the top 300 runs for each evaluation metric are reported in Table 2.5. Empirically, we observe that smaller values of k generally enhance attack performance, though this effect varies by model.

2.D Additional Figures

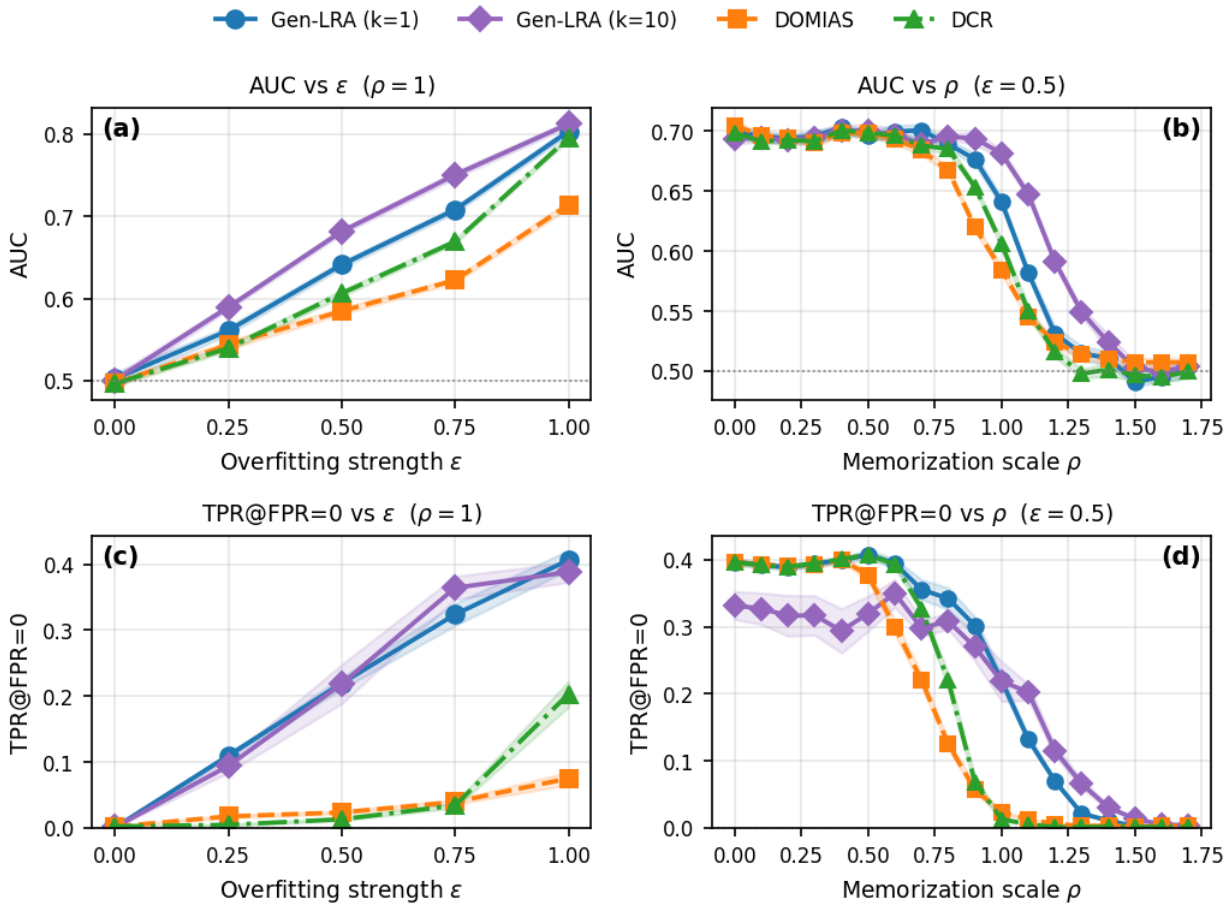


Figure 2.2: Simulation results across attacks and metrics. **(a, c)** AUC and TPR@FPR=0 versus ϵ at $\rho = 1$. All attacks scale linearly with ϵ; Gen-LRA achieves the largest slope on both metrics. **(b, d)** AUC and TPR@FPR=0 versus ρ at $\epsilon = 0.5$. At small ρ all attacks perform comparably; at moderate ρ Gen-LRA dominates while baselines collapse; at large ρ all attacks approach random. The locality parameter k controls how Gen-LRA aggregates evidence across synthetic neighbors: $k = 1$ recovers the full TPR@FPR=0 signal under near-exact memorization, where evidence concentrates on a single neighbor, while $k = 10$ outperforms it at moderate ρ, where the signal is dispersed across multiple neighbors.

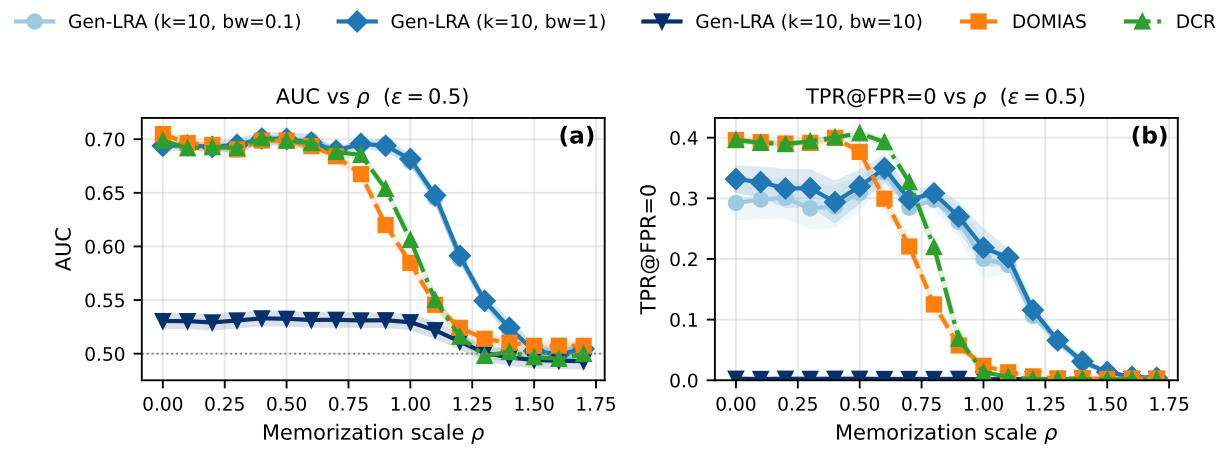


Figure 2.3: AUC and TPR@FPR=0 versus memorization scale ρ for Gen-LRA $k = 10$ at three bandwidths (0.1×, 1×, and 10× Silverman's rule), with DOMIAS and DCR shown for reference. Narrow and default bandwidths produce nearly indistinguishable curves; the wide bandwidth (10×) oversmooths the perturbation from $x^\star$ and collapses attack performance to near-random.

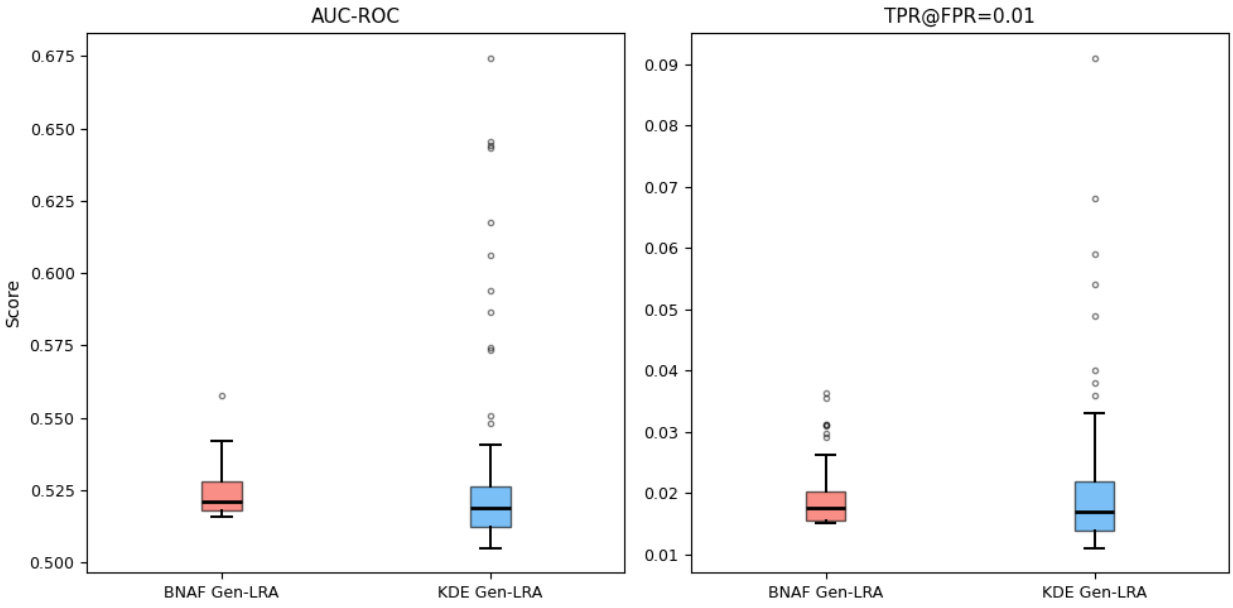


Figure 2.4: Distribution of Gen-LRA performance across benchmark runs using BNAF and KDE as the surrogate density estimator, on AUC-ROC (left) and TPR@FPR=0.01 (right). BNAF produces a tighter distribution of scores but fails to detect the strongest leakage signals: its upper tail is bounded well below KDE's, whose long upper tail captures the runs on which the most vulnerable generators leak training-set membership.

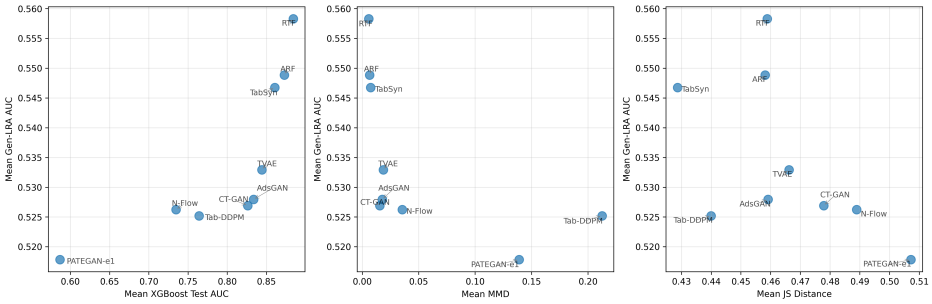


Figure 2.5: Mean Gen-LRA AUC plotted against three quality measures across generators: downstream utility (XGBoost test AUC under train-on-synthetic-test-on-real), maximum mean discrepancy, and Jensen-Shannon distance.

Chapter 3

Finding Connections: Membership Inference Attacks for the Multi-Table Synthetic Data Setting

3.1 Introduction

Synthetic tabular data has been shown to enable the sharing of sensitive or private information [1, 2]. While considerable progress in synthetic data generation has focused on single table applications, where a generative model learns the distribution of a single table, most real world data exists in hierarchical structures stored in relational databases, where rows in one table have interdependencies with rows in other tables [61–63]. Modeling and producing synthetic databases rather than single tables has seen growing interest as it allows for the release of more expressive data and lately a variety of methods have been developed to learn and generate synthetic databases [64–66, 29, 67, 68].

While the results of these models are impressive, auditing the empirical privacy of synthetic database release is not well understood. Unlike single table settings that protect privacy at a row or item level, in relational databases a user's information is distributed across items in multiple interconnected tables. As we

The contents of this chapter appeared in: Ward, J. et al., "Finding Connections: Membership Inference Attacks for the Multi-Table Synthetic Data Setting," 2026.

will show in Sections 3.3.5 and 3.5, privacy leakage that occurs over a particular item implies the leakage of all connected items.

Membership Inference Attacks (MIAs) have been successfully applied to audit the privacy of single table tabular generators and can be used to estimate the empirical ϵ differential privacy of generated synthetic data [30, 14]. However, we demonstrate that single table MIAs are inadequate for auditing multi table synthetic data privacy at the user level because they can only audit privacy at the item level. Current attacks, while applicable to item level privacy auditing in individual tables within a synthetic database, fail to exploit the critical inter tabular relationships that define a user across multiple tables, rendering them unsuitable for comprehensive user level privacy auditing in relational database contexts.

To our knowledge the first work to study this problem, we propose a novel Membership Inference Attack setting designed to audit the *user level privacy* of multi table synthetic data release. Here, we construct relational databases as heterogeneous graphs in which we infer if a test *subgraph* describing all connected information for a user was included in the training set of the generative model that produced the synthetic database. We then propose a new attack, Multi Table MIA (MT-MIA), which performs membership inference by learning graphical representations of user centric subgraphs using heterogeneous graph neural networks (HGNNs) [69–74] under a No-Box threat model.

Unlike existing tabular MIAs, MT-MIA leverages all relational information associated with a user, explicitly targeting inter-table conditional dependencies that are inaccessible to single table formulations. The attack is model agnostic and can be applied to arbitrary multi table datasets and synthetic data generators, making it broadly applicable to both practitioners and researchers.

To validate MT-MIA, we construct examples of multi table privacy leakage and show that current single table, item level attacks are no better than random guessing at user level membership inference in these scenarios whereas MT-MIA achieves near perfect AUC, highlighting the need for user level specific privacy auditing techniques. We then empirically deploy MT-MIA on a variety of real world multi table datasets, finding that current state of the art multi table generators possess this unique vulnerability, even under a conservative threat model. Finally, we analyze the intermediate embedding spaces of MT-MIA to show that MT-MIA can diagnose where memorization may be occurring in the multi table training set. Overall, these results demonstrate that effective privacy auditing for multi table generative models requires user level analyses, and that MT-MIA provides a practical mechanism for uncovering such leakage in real world settings.

3.2 Related Works

3.2.1 Synthetic Data Generation and Release

Non-Relational Tabular Data: The objective of synthetic data generation is to learn the probability distribution of a training dataset from which to generate new artificial samples that exhibit statistical properties similar to the original data. In the non-relational tabular data case, each row typically represents a complete entity (e.g., a single user with all their associated attributes), with each entity modeled as an independent observation or item associated with a fixed set of features. Techniques such as generative adversarial networks [38, 1, 2], language models [4, 29], and diffusion models [3, 75, 54], have demonstrated an impressive ability to generate realistic and diverse synthetic data.

Relational Tabular Data: While single-table approaches are useful in certain applications, most data of release interest in healthcare, education, finance, and government are not stored as isolated tables but rather in relational databases [62, 72, 63].

Definition 2 (Relational database)**.** A relational database $\mathcal{D} = (\mathcal{T}, \mathcal{J})$ consists of a collection of tables $\mathcal{T} = \{T_1, \ldots, T_n\}$ and joins or links between these tables $\mathcal{J} \subseteq \mathcal{T} \times \mathcal{T}$. Each table is a set $T' = \{o_1, \ldots, o_n\}$ where the elements $o \in \mathcal{O}$ are referred to as rows or observations. Each observation is a tuple $o = (\mathcal{P}_o, \mathcal{K}_o, f_o)$ where:

- $\mathcal{P}_o$ is the Primary Key that uniquely identifies the observation o.
- $\mathcal{K}_o$ is the set of Foreign Keys corresponding to a primary key in other tables, thus connecting the tables.
- f_o corresponds to the features or columns of the observation o.

In contrast to the single-table case, relational tabular data distributes information about a single user or entity across items in multiple tables connected through conditional joint relationships. As relational tabular data can be more expressive than its non-relational counterpart, a number of methods have been proposed to learn and generate synthetic relational data using probabilistic methods [64, 66], transformers [65], latent diffusion [67], and language models [29]. A common theme across these approaches is to define chains of modular generators for each table based on the parent-child relationships implied by the join relationships in a database schema. The typical generation process begins by synthesizing data for parent tables, then recursively and conditionally generating their children while controlling cardinality and join relationships through histogram-based or clustering-based mechanisms.

The release of these data raises unique privacy challenges compared to the non-relational setting as the *granularity* of the unit of privacy the releasing party wishes to protect can change. In single-table scenarios, one independent entity or user is a row or item. In contrast, relational data structures represent a user as a set of inter-related items. This interconnectedness means that protecting privacy is no longer confined to an *item-level* but rather a *user-level*. As we will show in Section 3.3, leakage of an item or join relationship implies privacy leakage about all related rows for a user, and current item-level auditing procedures underestimate this risk.

3.2.2 Membership Inference Attacks for Synthetic Data Generation

MIAs are a class of adversarial techniques that aim to distinguish between member records—those used in the training set of a target model—and nonmember records—those drawn from an independent dataset [9]. Originally proposed in the context of classification models [32, 41, 11, 42, 31, 76], MIAs exploit the tendency of learning algorithms to behave differently on training data than on unseen data, often due to overfitting or memorization. As a result, MIAs have become a central tool for empirically auditing privacy leakage in machine learning systems, complementing formal guarantees such as differential privacy. More recently, MIAs have been adapted to the setting of tabular synthetic data generation, where the adversary's goal is to infer whether a record contributed to the training of a generative model based on access to synthetic samples. In this context, successful membership inference indicates that the synthetic data preserves information too faithfully, potentially enabling privacy violations even when direct record linkage is not possible.

Single-Table MIAs

Membership inference attacks against single-table synthetic data aim to determine whether a given real record influenced the training of a generative model, based solely on properties of the released synthetic dataset and, in some cases, limited auxiliary information. Unlike MIAs for discriminative models, where the attacker probes a target model's outputs directly, attacks in the synthetic data setting must infer membership indirectly through distributional artifacts left behind by the generation process.

A broad class of attacks operates using only the released synthetic table and, in some cases, auxiliary reference data, without access to the generative model. These attacks exploit the observation that records drawn from the training set often induce locally atypical behavior in the synthetic distribution, such as elevated neighborhood density, reduced variability, or near-duplicate synthetic samples. Operationally, they measure the extent a candidate record is memorized or overfit too using using distances, density estimates, or reconstruction scores to classify record membership [44, 45, 10, 13, 30, 77]. By relying only on released

samples and optional auxiliary data, these methods avoid assumptions about access to the generative model. Consequently, they are typically classified as operating in a no-box threat model. This setting is particularly relevant for private data release, where a data curator is unlikely to disclose generative model information to a potential adversary.

Stronger attacks assume access to additional information, most commonly partial or query access to the generative model. These attacks explicitly compare how likely a target record is under competing member and non-member hypotheses, often by training multiple shadow generative models to approximate each hypothesis and learning a decision rule to distinguish between them [14–16]. While such attacks can substantially improve inference accuracy, they are typically extremely computationally expensive, requiring repeated model training or large numbers of model queries. As a result, their practical relevance to synthetic data release is less clear, since data curators generally do not expose generative models or provide the level of access required to support such attacks.

Across this literature, attacks are formulated to audit item-level privacy for non-relational, single-table generative models, implicitly assuming that each record corresponds to an independent, fixed-dimensional feature vector. This assumption enables distance and density-based attacks and ties attack effectiveness directly to the degree of over-representation or memorization of individual training records in the synthetic data. In contrast, multi-table synthetic data induces a hierarchical or relational representation in which membership corresponds to the presence of a training entity across multiple linked tables. Attacks in this setting must implicitly learn a representation over sets of rows or relational subgraphs, rather than operating on a single row input.

Multi-Table MIAs

Compared to the single-table setting, membership inference attacks for multi-table synthetic data remain relatively underexplored. To date, there has been limited effort to formalize a threat model or define a standard MIA setting of entity membership across multiple relational tables.

The primary empirical study in this setting is the MIDST Competition at SATML 2025 [78], which evaluated membership inference attacks against multi-table synthetic data generated by ClavaDDPM under a range of white-box and black-box threat models. Despite access to multiple linked synthetic tables, the competition hosts noted that the strongest attacks relied exclusively on a designated main table, effectively reducing the problem to a single-table setting. Indeed, the winner of the competition ignored auxiliary tables entirely [79]. Attacks that attempted to incorporate information from auxiliary tables or relational structure were noted to not yield improved performance and that they often performed worse than single-table

strategies.

Overall, existing empirical results suggest that current membership inference techniques do not successfully exploit relational dependencies in multi-table synthetic data, and that effective attacks in this setting have yet to be demonstrated.

3.2.3 Heterogeneous Graph Neural Networks

Heterogeneous Graph Neural Networks (HGNNs) provide the necessary inductive bias to preserve the multi-modal semantics of relational databases by explicitly modeling the distinct node and edge types inherent in database schemas. Unlike homogeneous GNNs, which treat all connections as semantically equivalent, HGNNs utilize type-specific transformation matrices and message-passing protocols to navigate the "web" of relational tables [80]. This capability is essential for membership inference in multi-table contexts, as privacy leakage often resides not in a single row, but in the specific structural alignment between a parent entity and its conditionally generated child records.

Early architectures in this domain, such as the Heterogeneous Graph Attention Network [70], relied on hierarchical attention mechanisms—operating at both the node and semantic levels—to aggregate information across predefined meta-paths. However, the requirement for manual path engineering often limits their flexibility in complex database schemas. To address this, more recent frameworks like the Heterogeneous Graph Transformer [81] and the Graph Attention Transformer operator [82] introduce dynamic, typed-attention mechanisms that automatically learn the importance of different relational dependencies.

The recent formalization of Relational Deep Learning [63] further validates the use of HGNNs for database-centric tasks, demonstrating that structural representations can effectively capture the joint distributions of tabular data without the loss of information inherent in table flattening. MT-MIA leverages these advancements to perform privacy auditing, utilizing HGNNs to detect "memorized motifs"—instances where a synthetic generator reproduces training structures.

3.3 A User-Level Membership Inference Attack Setting

Releasing relational synthetic data implies a notion of privacy at the user-level which differs from the item-level privacy implied in the single-table setting. In this section, we develop a novel MIA setting to audit the empirical privacy of user entities and show that item-level privacy does not necessarily provide privacy protection at the user level, necessitating MT-MIA, a new membership inference technique proposed in Section 3.4 that follows this setting.

3.3.1 Relational Databases as Heterogeneous Graphs

We represent a relational database $\mathcal{D}$ as a heterogeneous graph in order to reason about user level entities as connected subgraphs rather than isolated rows. This representation makes cross table dependencies explicit and is what allows our attack to operate on the full set of records belonging to a user.

Definition 3 (Database as a heterogeneous graph)**.** Given a database $\mathcal{D} = (\mathcal{T}, \mathcal{J})$, we construct a heterogeneous graph $G = (\mathcal{V}, \mathcal{E}, \mathcal{T}_V, \mathcal{T}_E, X_V)$ as follows:

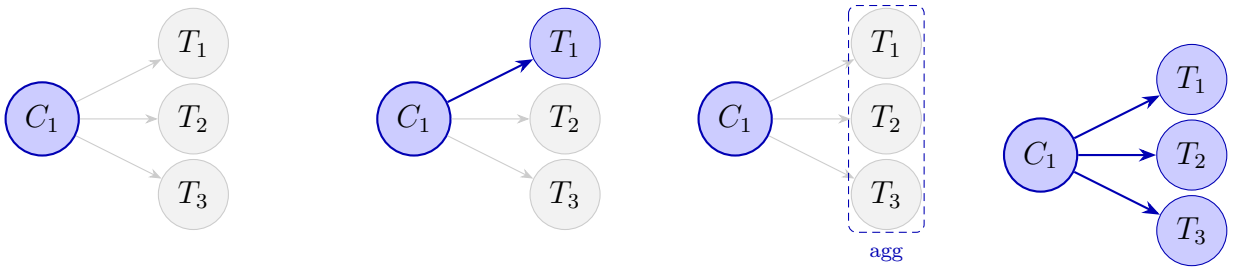
- Each table $T_i \in \mathcal{T}$ defines a node type $t_i \in \mathcal{T}_V$. Each row $o \in T_i$ becomes a node $v \in \mathcal{V}$, with type assigned by $\phi_V : \mathcal{V} \to \mathcal{T}_V$, $\phi_V(v) = t_i$.
- The non key features f_o of a row form the node's attribute vector $x_v \in X_V$.
- Each Foreign Key relation in $\mathcal{J}$ defines an edge type $r \in \mathcal{T}_E$. A Foreign Key reference from row u to row v instantiates a directed, typed edge $e = (u, r, v) \in \mathcal{E}$.

The schema is preserved as a typed meta structure: the set of admissible $(\phi_V(u), r, \phi_V(v))$ triples is exactly the set of Foreign Key relations in $\mathcal{D}$. A user entity, a set of rows transitively linked via Foreign Key relations, corresponds to a connected, typed subgraph of G. This graph perspective is particularly valuable for our membership inference context, as it enables us to trace information leakage across table boundaries and identify how patterns in relational data might reveal user membership despite protections that may be effective at the individual table level. For the remainder of the paper, we use graph language interchangeably with database language: nodes refer to rows, node types to tables, edges to Foreign Key references, and edge types to Foreign Key relations.

3.3.2 Formalism

In the multi table synthetic data generation setting, a generative model $\mathcal{M}$ is trained on a heterogeneous graph $G_{\text{train}} \sim \mathbb{P}$ that is a random sample of the population. $\mathcal{M}$ is then sampled to produce a synthetic graph G_{synth}. We define an entity subgraph h as a connected subgraph of G corresponding to all rows associated with a single user. Following the classical Membership Inference game [9], we define a test entity subgraph h^* as either a subgraph of G_{train} or a fresh sample from a holdout graph $G_{\text{holdout}} \sim \mathbb{P}$. Let $\mathcal{H}$ denote the set of all such test entity subgraphs. An adversary $\mathcal{A} : \mathcal{H} \to \{0, 1\}$, with access to G_{synth} and perhaps other information defined by a threat model, aims to determine the membership for a given h^*. Formally, this Membership Inference Attack can be expressed as:

$$\mathcal{A}(h^*) \;=\; \mathbb{I}\{f(h^*) > \gamma\}\,, \tag{3.1}$$



(a) Customer Features Only

(b) Join to Single Transaction

(c) Transaction Aggregation

(d) MT-MIA (Ours)

Figure 3.1: Comparison of attack strategies for Section 3.3.5. All methods operate on the same Customer-Transaction graph (C_1 with three transactions T_1, T_2, T_3). Blue highlighting indicates information used by each attack; gray indicates ignored or collapsed information. Single-table based attacks have three options in constructing their attack: **(a)** Use only customer features, ignoring all relational information. **(b)** Join each customer to a single transaction, requiring arbitrary sampling and discarding remaining relationships. **(c)** Joining customers to arbitrarily aggregated transaction features, collapsing the relationship structure. **(d)** MT-MIA preserves and learns from the complete graph structure, automatically discovering that relationship cardinality reveals membership (AUC = 0.999).

where $\mathbb{I}$ is the indicator function, f is a scoring function evaluated on h^* and G_{synth}, 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 the privacy leakage introduced by the release of G_{synth} sampled from the model trained on the original data.

This formulation tests whether $h^* \subseteq G_{\text{train}}$. It differs from traditional MIAs that test if an independent item or row was included in the training dataset; here, we evaluate whether the generative model preserves the privacy of all related nodes for a user $v \in h^*$ when released together.

Example: Consider a relational database with two tables: `Customers` and `Transactions`, where each customer may place multiple distinct transactions (one-to-many relationship). This schema induces a heterogeneous graph in which each customer node connects through edges to a set of transaction nodes, forming a connected subgraph for each customer. In this setting, our task is to infer whether such a subgraph, representing a distinct customer and their full transaction history, was used for training.

3.3.3 Privacy Auditing with Subgraphs

In theory, h^* could be any subgraph of interest to an adversary or auditor, since if $h^* \subseteq G_{\text{train}}$, then $\forall g \subseteq h^*, g \subseteq G_{\text{train}}$. However, it is useful to add several conditions to simplify auditing procedures, as

conducting membership inference on all possible constructions of h^* is often computationally infeasible. First, we restrict h^* to be a connected subgraph, as by the construction of relational databases two unconnected nodes imply independence. Second, many real world databases naturally decompose into disjoint connected subgraphs, where each subgraph represents a unit or entity, such as a customer and their associated transactions. We thus propose auditing a finite, well structured set of subgraphs by leveraging the relational decomposition of databases.

Let $H_{\text{test}} = \{h_1, h_2, \ldots, h_n\}$ denote the set of all disjoint connected entity subgraphs in $G_{\text{test}} = G_{\text{train}} \cup G_{\text{holdout}}$, where $h_i \subseteq G_{\text{test}}$ and $h_i \cap h_j = \emptyset$ for $i \neq j$. We propose to audit membership specifically over H_{test}. This formulation and these assumptions lead to the following result:

Theorem 4. *Let $G_{test} = (V, E)$ be a graph that is the disjoint union of two subgraphs G_{train} and $G_{holdout}$, where $V(G_{test}) = V(G_{train}) \cup V(G_{holdout})$ and $V(G_{train}) \cap V(G_{holdout}) = \emptyset$. Furthermore, there are no edges in G connecting vertices between G_{train} and $G_{holdout}$ (i.e., all edges in G have both endpoints in either G_{train} or in $G_{holdout}$). Let $h^* \subseteq G$ be a connected subgraph, and let $g \subseteq h^*$.*

Then, if $g \subseteq G_{train}$, it follows that $h^* \subseteq G_{train}$. Likewise, if $g \subseteq G_{holdout}$, then $h^* \subseteq G_{holdout}$.

A proof is included in Appendix 3.A. As a sketch, by construction h^* cannot have nodes nor edges that connect to nodes in both G_{train} and G_{holdout}. This exclusivity establishes that the membership of all nodes in g and h^* cannot differ. An immediate corollary is that under the disjointness assumption, $h^* \subseteq G_{\text{train}}$ and $h^* \subseteq G_{\text{holdout}}$ are mutually exclusive: every candidate subgraph in H_{test} has a well defined membership label.

We note that this auditing setup aligns with how multi table synthetic data generators are trained in practice. Generators must be fit on entity subgraphs that preserve the full set of rows belonging to each user, because the joint distribution of a user across tables is precisely what they are designed to model; partial subgraphs would distort the conditional dependencies between parent and child rows and yield a generator that is unfaithful to the source distribution. Membership at the level of complete entity subgraphs is therefore the natural unit of inference for relational synthetic data release.

Consequence of Multi Table Synthetic Data Release. While the assumption of Theorem 4 is not strictly required for relational data MIAs, it emphasizes the privacy risk of relational synthetic data release: if privacy leakage occurs over any component of a connected subgraph, it implies that the entire subgraph (and all included nodes) must be a member of the same source graph. In other words, it is not enough to protect the privacy of the observations of one individual table, as *all connected information can risk membership inference.*

3.3.4 Threat Model

In this work, we explore membership inference attacks under a No-Box threat model. In the No-Box setting, the attacker has access only to a single synthetic dataset G_{synth} as well as a database schema and must reason about membership without any knowledge of the generator architecture, training procedure, or internal parameters. This threat model reflects scenarios where a party has published a synthetic dataset in isolation with no additional model implementation details, and an adversary must assess privacy risks based solely on patterns and statistical properties present in G_{synth}.

While a variety of other threat models have been studied for synthetic tabular data, including Calibrated No-Box where the adversary has an additional reference dataset [30, 77] and Shadow-Box where an adversary additionally has implementation knowledge of the generator in order to generate shadow models [14–16], No-Box is the threat model that most closely matches how synthetic relational data is released in practice. In typical deployments, a data curator publishes a synthetic dataset without releasing the generator, its training data, or any auxiliary reference data, and an auditor or adversary must reason about membership from the synthetic dataset alone. The other threat models impose assumptions that are implausible in the multi table setting:

- **Calibrated No-Box** assumes the adversary additionally holds a fresh sample from the same population distribution as the training data, which in the relational setting requires an entire reference database that faithfully reproduces the joint distribution over node attributes and edge relationships across all tables. This is a substantially stronger assumption than its single table analog and is implausible in the settings synthetic relational data release is meant to enable, such as healthcare and finance, where the whole reason to release synthetic data is that comparable real data is not available.
- **Shadow-Box** grants the adversary knowledge of the model implementation along with a reference dataset, enabling the construction of shadow models. This is even more implausible in relational synthetic data release: such attacks are trivially defeated by not publishing implementation details, and are computationally infeasible for modern relational generators. In our experimentation, a single training run of RelDiff took 48 hours on an H200 to converge under default hyperparameters.

While No-Box attacks represent a lower bound on the privacy leakage detectable under more powerful threat models, this setting is most operationally relevant as it is the most realistic adversarial setting for released synthetic relational dataset in practice.

Additionally, a No-Box threat model also allows MT-MIA to be both model agnostic and dataset agnostic, enabling straightforward auditing of newly proposed relational data generators as they are developed. Because the attack operates solely on the released synthetic data and schema, it does not require adaptation to

generator specific interfaces or assumptions, making it applicable in post hoc privacy evaluations where only the synthetic dataset is available.

3.3.5 Motivating Example: The Need for Multi-Table Attacks

To illustrate how inter-table dependencies can leak privacy, we construct a toy example (full experimental details are included in the Appendix). Consider a database with two tables: *Customers* and *Transactions*, connected by a one-to-many relationship where multiple transactions belong to a single customer. Both tables have identical feature distributions following $\mathcal{N}(0, I)$.

In our constructed scenario, non-member samples contain exactly one transaction per customer, while member samples (training data) contain 100 transactions per customer—a pattern that might arise from data drift or sampling bias. A synthetic data generator produces G_{synth} with the same distributional properties as the training set. Our goal is to infer membership of customer entities based solely on the release of G_{synth} under a No-box threat model.

From an adversarial perspective, this leakage is trivial to exploit: a decision rule that predicts membership based on whether a customer has 100 transactions achieves perfect accuracy. However, existing single-table MIAs applied to this scenario such as Distance to Closest Record [10] and MC [45] fail to detect this leakage. As shown in Figure 3.1, single-table approaches must make arbitrary choices about how to incorporate multi-table information. A practitioner using existing single-table MIAs would typically either (Figure 3.1a) ignore transaction data entirely , (Figure 3.1b) join each customer to a single arbitrarily-chosen transaction , or (Figure 3.1c) aggregate transaction features (e.g., computing means or sums). While aggregation approaches *could* capture this signal (such as counting the transactions), this requires apriori adversarial knowledge of the leakage as to which relational aspects matter—an assumption that does not scale to complex schemas or subtle leakage patterns.

In contrast (See Figure 3.2), MT-MIA automatically learns from the full relational structure (Figure 3.1d) and achieves an AUC of 0.999 without manual feature engineering. While this extreme example is unlikely in practice, it exemplifies a fundamental principle: **inter-table relationships can leak membership signal**, and user-level adversarial auditing must account for the full multi-table structure.

3.4 Methodology: MT-MIA

The motivating example in Section 3.3.5 demonstrates a fundamental vulnerability in tabular synthetic data auditing: *single table MIAs are topologically limited*. Even when a generator overfits to or memorizes

inter-table correlations or cardinality, single table attacks fail to capture this signal because they lack a mechanism to process non Euclidean relational dependencies.

To address this, we propose Multi Table Membership Inference Attack (MT-MIA). MT-MIA is designed to be *schema agnostic*, utilizing a heterogeneous graph encoder to map complex relational structures into a low dimensional embedding space. Instead of relying on manual feature engineering or arbitrary aggregation, MT-MIA leverages graph representation learning to identify discriminative structural patterns directly from G_{synth}.

The intuition for MT-MIA is that by incorporating all of an entity subgraph's information into the learned embedding space, the resulting embeddings will be more discriminative than any individual table's representation alone. We first describe the HGNN backbone that maps relational structures into a latent space, then explain how we train the model and score the membership of target subgraphs. Notation introduced throughout this section is summarized in Appendix 3.B.

3.4.1 Graph Encoder Architecture

The core of MT-MIA is a heterogeneous graph encoder $\mathcal{M}_\theta$, parameterized by learnable weights θ (see Figure 3.3). The HGNN architecture provides the *topological inductive bias* required to model relational dependencies. Unlike traditional autoencoders, $\mathcal{M}_\theta$ is invariant to specific join relationships as well as table and row cardinality, enabling a unified attack interface across any heterogeneous relational schema.

Heterogeneous Message Passing To capture the semantics of the relational schema, we stack L heterogeneous message passing layers. For a node type $t \in \mathcal{T}_V$ at layer l, the layer's node feature update is:

$$H_{l+1}^{(t)} = \Phi_l^{(t)}\left(H_l^{(t)}, \{A^{(r)}\}_{r \in \mathcal{T}_E}\right), \tag{3.2}$$

where $H_l^{(t)}$ is the matrix of layer l node embeddings for nodes of type t, $\Phi_l^{(t)}$ is a type specific message passing function, and $A^{(r)}$ is the adjacency matrix for relation type r. Throughout the paper, we instantiate $\Phi_l^{(t)}$ as GATv2 [82] due to its expressive attention mechanism. This mechanism enables information propagation both within and across node types by leveraging the typed edges in the heterogeneous subgraph.

Dynamic Gated Fusion While the bifurcated signals provide a comprehensive view of the entity, the discriminative properties of these embeddings are unknown to the adversary. Privacy leakage may manifest in the unconditionally generated attributes (z_{parent}), the conditionally generated relational dependencies (z_{context}), or a latent intersection of both. To address this uncertainty, we employ a *Dynamic Gating Unit*

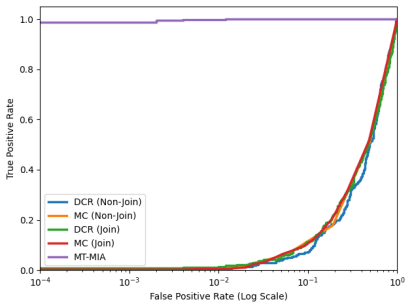


Figure 3.2: Plot of the True Positive Rate by log-scaled False Positive Rate for Section 3.3.5. Conventional single-table attacks Distance to Closest Record [10] and MC [45] cannot distinguish membership from the Customers table with or without additional Transaction rows joined as they cannot exploit the given inter-tabular leakage. MT-MIA learns and attacks a representation of the entire user subgraph allowing for nearly perfectly membership discrimination (AUC=0.999).

that adaptively modulates the integration of these signals.

We compute a learned gating vector $g \in [0,1]^d$ that serves as an entry wise modulator for relational influence. Given the parent and context embeddings, the gate is formulated as:

$$g = \sigma(\text{MLP}_{\text{gate}}([z_{\text{parent}} \parallel z_{\text{context}}])) \,, \tag{3.3}$$

where σ is the elementwise sigmoid activation and MLP_{gate} is a small multilayer perceptron applied to the concatenation $[z_{\text{parent}} \parallel z_{\text{context}}]$. The final composite representation z_{final} is then constructed via a gated residual connection:

$$z_{\text{final}} = z_{\text{parent}} + (g \odot \varphi(z_{\text{context}})), \tag{3.4}$$

where $\varphi(\cdot)$ is a non linear transformation and $\odot$ denotes the Hadamard (elementwise) product. This mechanism allows $\mathcal{M}_\theta$ to "tune" the attack's sensitivity: it can prioritize intrinsic parent features when relational context is sparse, or amplify the structural signal when the generator exhibits strong conditional leakage. By allowing the data to determine the optimal weight of each signal, the encoder remains robust across various generative architectures and database schemas.

3.4.2 Training in the No-Box Setting

In the No-Box setting, the adversary has access only to the synthetic output G_{synth} without any knowledge of the generator's architecture, parameters, or training data G_{train}. Prior work in the single table setting [10, 45] establishes that synthetic data generators tend to leave detectable traces of memorization in their

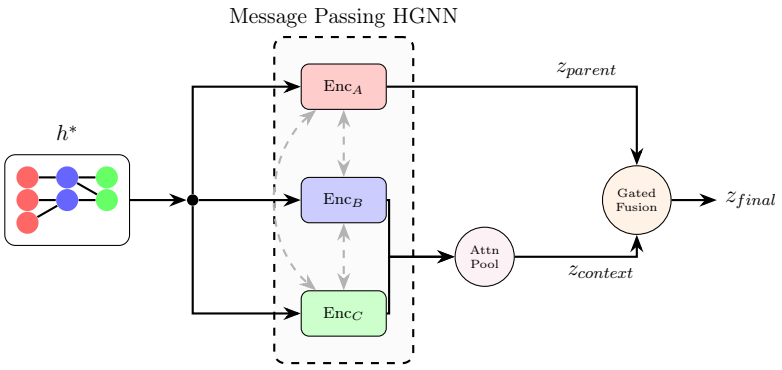


Figure 3.3: Inference Time Architecture Diagram of MT-MIA.

outputs: training records influence the synthetic distribution in ways that produce locally elevated density, near duplicate samples, or reduced reconstruction error around member records. Distance based attacks operationalize this observation by treating proximity between a candidate record and the synthetic dataset as evidence of membership. MT-MIA extends this principle to the relational setting: rather than measuring proximity in the raw feature space of a single row, we measure proximity in a learned embedding space that summarizes an entire entity subgraph, allowing the attack to detect memorization that manifests across connected rows.

Multi Anchor Reconstruction Objective Without access to ground truth membership labels, we train $\mathcal{M}_\theta$ using a self supervised reconstruction objective. The intuition is to force the encoder to learn a compressed representation that preserves both the primary entity's attributes and its relational neighborhood, which are the components that generators may inadvertently memorize during training.

We introduce two complementary reconstruction heads: a *parent decoder* $\mathcal{D}_\phi$ with parameters ϕ that reconstructs the parent node's features, and a *context decoder* $\mathcal{D}_\psi$ with parameters ψ that reconstructs the aggregate neighborhood. For a target node $i \in \mathcal{V}_t$ with neighbor set $\mathcal{N}(i)$ spanning all adjacent node types, the composite reconstruction loss is:

$$\mathcal{L}_{\text{recon}}(\theta, \phi, \psi) = \mathbb{E}_{i \sim \mathcal{V}_t} \Bigg[\lambda_p \underbrace{\| \mathcal{D}_\phi(\mathcal{M}_\theta(x_i)) - \mathbf{x}_i^{(t)} \|_2^2}_{\text{Parent Recon.}} + \lambda_c \underbrace{\Big\| \mathcal{D}_\psi(\mathcal{M}_\theta(x_i)) - \sum_{j \in \mathcal{N}(i)} \mathbf{x}_j \Big\|_2^2}_{\text{Context Recon.}} \Bigg], \tag{3.5}$$

where $\mathcal{M}_\theta(x_i)$ produces the fused embedding from Equation 3.4 and $\lambda_p, \lambda_c \in \mathbb{R}_{\geq 0}$ are hyperparameters that balance the reconstruction of parent features against relational context. By learning to reconstruct both signals from a single bottleneck embedding, the encoder is forced to capture the specific relational motifs and

conditional dependencies favored by the generator. Records exhibiting similar motifs produce embeddings that cluster near G_{synth} in this learned space.

Membership Scoring While the learned embedding space can support various No-Box attacks, we derive our membership scoring from the Distance to Closest Record (DCR) attack [10]. For a candidate entity subgraph h^*, we define $f(h^*)$ from Equation 3.1 as:

$$f(h^*) = -\min_{h \subseteq G_{\text{synth}}} \|\mathcal{M}_\theta(h^*) - \mathcal{M}_\theta(h)\|_2. \tag{3.6}$$

Higher scores indicate that the structural motifs of h^* were likely memorized and reproduced in G_{synth}.

3.5 Experiments

To evaluate the effectiveness of MT-MIA, we conduct a series of experiments on three benchmark multi-table datasets: California Census [83], Airline Customers [84], and Airbnb [85]. For each dataset, we begin by constructing training and holdout sets through the sampling of disjoint subgraphs, ensuring no overlap in entities or relationships. The synthetic data generator is then trained with the training subgraphs, after which we sample an equal number of synthetic subgraphs to match the original training size. All numeric features are scaled and categorical features are ordinally encoded in relation to the synthetic data, which are then applied consistently to both real and synthetic samples to prevent data leakage.

We experiment with a training size of 1000 user subgraphs. These subgraphs correspond to thousands of items across all datasets' tables. To account for randomness in model training and sampling, each experimental configuration is repeated across three independent runs. Following the recommendations of prior work [86], we fix the data split (training vs. holdout) across all runs and vary only the generative model initialization seeds. This design helps isolate the variability due to model behavior from that due to evaluation set construction, which is especially important in privacy attack scenarios.

Following [30], [87] and [13], all training data are included as the positive membership class with an equal sized holdout dataset as a negative class. All MIAs are then evaluated with the corresponding synthetic data on this evaluation set to then calculate the success of the attack.

We run all experiments on a High Performance Computing Cluster using a Nvidia H200 GPU with a 32 core CPU. The full synthetic data generation procedure for the models RealTabFormer and ClavaDDPM was approximately 10 hours of compute on this system. For the much larger and more expensive RelDiff, the procedure was approximately 450 GPU hours. The MT-MIA training and inference procedure was

approximately 1 hour of compute time over all runs. Additional compute was used for preliminary experiments.

3.5.1 Baselines

Multi-Table Synthetic Data Generators

We evaluate our proposed approach against three representative multi relational generative models that span autoregressive sequence modeling, hierarchical diffusion, and graph structured diffusion approaches.

- **RealTabFormer [29]:** This model synthesizes multi-relational data by framing child table generation as a conditional sequence generation task. It utilizes a GPT-based architecture where parent records are encoded to form a context for a sequence-to-sequence (Seq2Seq) model. The generator produces child rows while maintaining one-to-many relationship cardinality by treating primary-key-foreign-key links as causal sequences. This approach models conditional distributions without requiring manual schema flattening.
- **ClavaDDPM [67]:** This framework generates multi-relational data through a hierarchical guidance mechanism using cluster-based latent variables. The model applies a clustering algorithm to the parent table to extract latent representations, which serve as conditioning signals for the diffusion process of the associated child tables. During the reverse denoising step, the model optimizes a conditional objective function to align child records with parent clusters. This structure captures dependencies across the database schema without the computational overhead of autoregressive or graph-based methods.
- **RelDiff [68]:** This framework synthesizes complete relational databases by explicitly modeling their Foreign Key graph structure. RelDiff decomposes generation into two stages: a joint graph conditioned diffusion process that synthesizes attributes across all tables simultaneously, and a Stochastic Block Model based graph generator that synthesizes the Foreign Key structure itself. This decomposition of graph structure from relational attributes is designed to preserve both fidelity and referential integrity, avoiding the structural assumptions imposed by methods that flatten relational data into conditionally generated tables.

Attacks

Since prior work has not investigated MIAs in the context of synthetic database release, we adapt existing single-table attack methods that align with our threat model for benchmarking purposes. In our setting,

Table 3.1: Per-(model, dataset) comparison of MT-MIA against the best baseline attack. All values are means across seeds. For each pair, the larger value is **bold**; Δ reports the absolute gain of MT-MIA over the baseline. A table with standard deviations are reported in Appendix 3.D.

Model	Metric	California			Airbnb			Airlines		
		Baseline	MT-MIA	Δ	Baseline	MT-MIA	Δ	Baseline	MT-MIA	Δ
ClavaDDPM	AUC-ROC	**0.79**	0.69	−0.10	0.79	**0.80**	+0.01	**0.69**	0.66	−0.03
	TPR@FPR=0	0.00	**0.07**	+0.07	0.00	**0.01**	+0.01	0.07	**0.17**	+0.10
	TPR@FPR=10^{-3}	0.01	**0.09**	+0.08	0.01	0.01	0.00	0.08	**0.21**	+0.13
	TPR@FPR=10^{-2}	0.11	**0.15**	+0.04	0.06	**0.09**	+0.03	0.12	**0.31**	+0.19
RelDiff	AUC-ROC	**0.67**	0.64	−0.03	0.57	**0.62**	+0.05	**0.51**	0.49	−0.02
	TPR@FPR=0	0.00	**0.15**	+0.15	0.00	**0.03**	+0.03	0.00	0.00	0.00
	TPR@FPR=10^{-3}	0.01	**0.17**	+0.16	0.00	**0.03**	+0.03	0.00	0.00	0.00
	TPR@FPR=10^{-2}	0.13	**0.22**	+0.09	0.02	**0.06**	+0.04	0.01	0.01	0.00
RTF	AUC-ROC	**0.57**	0.52	−0.05	0.58	0.58	0.00	**0.54**	0.51	−0.03
	TPR@FPR=0	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.01	0.00
	TPR@FPR=10^{-3}	0.00	0.00	0.00	**0.01**	0.00	−0.01	**0.02**	0.01	−0.01
	TPR@FPR=10^{-2}	0.02	0.02	0.00	0.02	0.02	0.00	**0.05**	0.03	−0.02

membership inference on a user subgraph h^* is equivalent to determining whether all nodes $v \in h^*$ were present in the training data. Per Theorem 4, this can be reframed as selecting 1 node for each subgraph in which to use a single-table attack on. We therefore select a parent node in each h^* (the strategy of Figure 3.1a) and use its corresponding score as the attack score for h^*.

We use three common No-box attacks for tabular data as these scoring functions: Distance to Closest Record (DCR) [10], a Monte Carlo Density Estimator (MC) proposed by [45], and a Kernel Density Estimator (KDE) approach used in [15].

3.5.2 Metrics

Attack Success

We evaluate attack effectiveness following the framework established by [11], which argues that membership inference attacks should be assessed by their behavior in the high confidence regime rather than by aggregate classification metrics. Our primary metric is the True Positive Rate at low False Positive Rates (TPR@FPR), reported at FPR levels of 0, 10^{-3}, and 10^{-2}. We also report the Area Under the ROC Curve (AUC) for completeness and comparability with prior work, but treat it as a secondary measure.

A high TPR at a low FPR is the most relevant regime for privacy auditing. An adversary that can confidently identify even a small fraction of training records with few false positives poses a real privacy risk, while an adversary that must accept many false positives to flag the same number of true members cannot reliably attribute leakage to any specific record. Attacks that only separate members and non members on average, across the full operating curve, do not produce this kind of high confidence identification.

Synthetic Data Fidelity

We evaluate the quality of the multi table synthetic data using four metrics : (1) *cardinality*, which measures the Foreign Key group size distribution to assess intra group correlations; (2) *column wise density estimation (1 way)*, which estimates the marginal density of every column across all tables; (3) *pairwise column correlation (k hop)*, which assesses dependencies between columns at distance k (for example, 0 hop for intra table and 1 hop for parent child relations); and (4) *average k hop*, which averages the k hop correlation scores over $k \in \{0, 1, \ldots, K\}$ for a schema of maximum join depth K, summarizing both intra table ($k = 0$) and cross table ($k \geq 1$) dependencies in a single fidelity score. For each measure following [67], we report the complement of the Kolmogorov Smirnov (KS) statistic and Total Variation (TV) distance, normalized to $[0, 1]$ where 1 indicates perfect fidelity.

3.6 Discussion

3.6.1 Privacy Auditing Multi-Table Synthetic Data

Performance of MIAs

MT-MIA's central advantage is that it scores membership over the full entity subgraph rather than a single row, which allows it to surface privacy violations that single table attacks are structurally incapable of detecting. The clearest evidence for this advantage appears in the high confidence regime that Section 3.5.2 identifies as the privacy relevant operating point of an MIA. Table 3.1 compares the mean best single table attack for each metric against MT-MIA. We find that MT-MIA delivers consistent and often substantial gains in TPR at low FPR across both ClavaDDPM and RelDiff.

The most striking pattern across our results is that MT-MIA reveals leakage where baselines detect none at all. On RelDiff with California, the strongest single table baseline achieves TPR@FPR=0 of 0.00, while MT-MIA reaches 0.15, a 15 percentage point gap that corresponds to confidently identifying roughly 150 training records with zero false positives in our evaluation. The same pattern holds on ClavaDDPM with California (0.00 to 0.07), ClavaDDPM with Airbnb (0.00 to 0.01), and ClavaDDPM with Airlines, where MT-MIA improves on the strongest baseline by 10 percentage points at FPR=0 and by 19 points at FPR=10^{-2}. In each of these settings, a practitioner relying on a single table audit would conclude that no high confidence privacy leakage is present; MT-MIA shows that this conclusion is wrong.

A notable pattern in these results is that incorporating relational data does not always improve AUC, yet consistently improves calibration in the high confidence regime. On several configurations, single table

Table 3.2: Fidelity metrics (left) versus MT-MIA privacy leakage (right) for each (model, dataset). Values are mean$_{\pm\sigma}$ over three seeds. Higher fidelity values indicate greater quality synthetic data; higher MT-MIA values indicate greater privacy leakage. RTF has substantially worse fidelity but this also translates to less privacy leakage, while ClavaDDPM and RelDiff achieve strong fidelity and leak substantially under MT-MIA.

Model	Dataset	Fidelity			MT-MIA	
		Col. Shapes	Cardinality	Avg. k-hop	AUC-ROC	TPR@FPR=10^{-3}
ClavaDDPM	California	$0.97_{\pm 0.00}$	$0.98_{\pm 0.01}$	$0.93_{\pm 0.01}$	$0.69_{\pm 0.05}$	$0.09_{\pm 0.08}$
	Airbnb	$0.98_{\pm 0.00}$	$0.98_{\pm 0.01}$	$0.92_{\pm 0.03}$	$0.80_{\pm 0.01}$	$0.01_{\pm 0.00}$
	Airlines	$0.99_{\pm 0.00}$	$0.99_{\pm 0.00}$	$0.97_{\pm 0.00}$	$0.66_{\pm 0.01}$	$0.21_{\pm 0.06}$
RelDiff	California	$0.97_{\pm 0.00}$	$1.00_{\pm 0.00}$	$0.93_{\pm 0.00}$	$0.64_{\pm 0.01}$	$0.17_{\pm 0.01}$
	Airbnb	$0.98_{\pm 0.00}$	$1.00_{\pm 0.00}$	$0.90_{\pm 0.03}$	$0.62_{\pm 0.01}$	$0.03_{\pm 0.02}$
	Airlines	$0.95_{\pm 0.01}$	$1.00_{\pm 0.00}$	$0.93_{\pm 0.01}$	$0.49_{\pm 0.01}$	$0.00_{\pm 0.00}$
RTF	California	$0.84_{\pm 0.01}$	$0.78_{\pm 0.13}$	$0.72_{\pm 0.01}$	$0.52_{\pm 0.02}$	$0.00_{\pm 0.00}$
	Airbnb	$0.90_{\pm 0.02}$	$0.85_{\pm 0.01}$	$0.75_{\pm 0.04}$	$0.58_{\pm 0.00}$	$0.00_{\pm 0.00}$
	Airlines	$0.87_{\pm 0.01}$	$0.06_{\pm 0.00}$	$0.80_{\pm 0.01}$	$0.51_{\pm 0.01}$	$0.01_{\pm 0.01}$

attacks achieve comparable or slightly higher AUC than MT-MIA while MT-MIA still detects strictly more leakage at low FPR. This suggests that the relational signal sharpens the high confidence end of the score distribution rather than uniformly separating members from non members.

A consistent property of MT-MIA across all three generators is that its scores track the inter-table signal each generator preserves. RealTabFormer, whose synthetic outputs lose substantial inter-table structure (Section 3.6.1), produces small MT-MIA gains over single table baselines because the relational motifs MT-MIA targets are largely absent from G_{synth}. ClavaDDPM and RelDiff, which preserve relational structure faithfully, are where MT-MIA surfaces large gains. The attack is therefore well calibrated to the property it is designed to detect: it returns strong signal where inter-table leakage exists in the synthetic output and stays muted where it does not.

Privacy versus Fidelity Tradeoff

We compare synthetic data fidelity against MT-MIA's detected leakage across all three generators in Table 3.2. ClavaDDPM and RelDiff achieve uniformly strong fidelity, with column shape, cardinality, and average k hop scores at or above 0.90 in nearly every cell. RealTabFormer's fidelity is substantially weaker for all metrics and datasets, particularly on Airlines where its cardinality score collapses to 0.06.

This fidelity gap maps directly onto leakage. Both high fidelity generators leak under MT-MIA: ClavaDDPM reaches TPR@FPR=10^{-3} of 0.21 on Airlines, and RelDiff reaches 0.17 on California. RealTabFormer leaks substantially less, with TPR@FPR=10^{-3} at or below 0.01 across all configurations. The relationship between fidelity and leakage is consistent with findings in the single table synthetic data literature: stronger preservation of intra and inter-table dependencies is correlated with greater privacy vulnerability [88, 87], and diffusion based generators in particular have been observed to be more susceptible to memorization based

Table 3.3: MIA metrics for intermediate and final embeddings found in MT-MIA for ClavaDDPM. We conduct DCR attacks on the single-table setting (Vanilla), the intermediate embeddings of MT-MIA z_{parent} and $z_{context}$, and the final embeddings in the attack z_{final}. ClavaDDPM experiences privacy leakage for different datasets in different components of multi-table synthesis.

Dataset	Metric	Vanilla	z_{parent}	$z_{context}$	z_{final}
California	AUC	**0.781**	0.768	0.650	0.746
	TPR@FPR=0	0.000	0.000	**0.183**	0.028
	TPR@FPR=10^{-3}	0.009	0.007	**0.282**	0.051
	TPR@FPR=10^{-2}	0.093	0.072	**0.315**	0.150
Airbnb	AUC	0.781	0.793	0.533	**0.795**
	TPR@FPR=0	0.000	0.000	0.000	**0.002**
	TPR@FPR=10^{-3}	0.006	0.006	0.001	**0.009**
	TPR@FPR=10^{-2}	0.056	0.061	0.013	**0.094**
Airlines	AUC	0.690	**0.713**	0.503	0.660
	TPR@FPR=0	0.055	**0.356**	0.002	0.155
	TPR@FPR=10^{-3}	0.089	**0.365**	0.007	0.245
	TPR@FPR=10^{-2}	0.122	**0.411**	0.014	0.308

attacks than alternative architectures. Our results indicate that this relationship extends to the multi table setting and applies to both diffusion based architectures we evaluate.

The interpretation of RealTabFormer's low MT-MIA scores warrants care. Table 3.1 shows that single table baselines do detect parent table leakage on RealTabFormer (AUC=0.57 on California, 0.58 on Airbnb). What MT-MIA finds less of in RealTabFormer's outputs is faithful inter-table structure: with cardinality and average k hop scores substantially below ClavaDDPM and RelDiff, the relational motifs MT-MIA scores against are likely not present in G_{synth} for the attack to exploit. MT-MIA's effectiveness is therefore tied to the strength of the inter-table signal a generator preserves: the more faithfully a generator reproduces relational structure, the more calibrated MT-MIA becomes relative to single table baselines.

3.6.2 Sources of Leakage: Node Versus Neighborhood

MT-MIA's performance gains stem from its ability to expose distinct sources of privacy leakage that are inaccessible to single table attacks. The HGNN backbone produces three intermediate representations of an entity subgraph: the parent embedding z_{parent}, the relational context embedding z_{context}, and the fused embedding z_{final}. This decomposition allows us to probe which components of the relational structure contribute to membership distinguishability and to attribute MT-MIA's gains to specific leakage pathways rather than to increased model capacity alone.

To quantify the contribution of each component, we apply the same DCR attack independently to each embedding space and compare against a Vanilla single table DCR attack on the original parent table feature space. We report attack performance for the best runs on each dataset under ClavaDDPM in Table 3.3.

On California, attacking z_{parent} yields AUC and TPR@FPR values nearly identical to the Vanilla attack, indicating that parent level attributes alone do not provide a substantially stronger signal once embedded.

Table 3.4: Child table attack comparison for ClavaDDPM (Mean ± Std). We deploy a single-table DCR against the children tables of each dataset and compare it to corresponding scores from MT-MIA targeting user graphs, exhibiting the "weakest link" effect implied by multi-table synthetic data release.

Dataset	Attack	AUC	TPR@FPR0	TPR@FPR10^{-3}	TPR@FPR10^{-2}
Airbnb	DCR	0.51±0.00	0.00±0.00	0.00±0.00	0.01±0.00
	MT-MIA	**0.79**±0.01	0.00±0.01	0.00±0.01	**0.09**±0.01
Airlines	DCR	0.51±0.00	0.00±0.00	0.00±0.00	0.01±0.00
	MT-MIA	**0.66**±0.02	**0.12**±0.08	0.12±0.08	**0.21**±0.13
California	DCR	**0.75**±0.00	0.00±0.00	0.00±0.00	0.04±0.00
	MT-MIA	0.69±0.05	**0.04**±0.04	**0.06**±0.05	**0.13**±0.11

In contrast, z_{context} reveals a markedly stronger signal, with improvements of 18 percentage points at TPR@FPR=0 and 28 percentage points at TPR@FPR=10^{-3} over Vanilla. This suggests that ClavaDDPM memorizes recurring relational motifs in child records, which remain invisible to single table attacks but become exploitable once relational neighborhoods are explicitly encoded.

The pattern inverts on Airlines and Airbnb, where the primary gains arise from z_{parent} rather than z_{context}. On Airlines, z_{parent} achieves a 30 percentage point increase in TPR@FPR=0 over Vanilla while z_{context} uncovers little to no membership signal. On Airbnb the effect is more modest: z_{parent} and z_{final} both outperform Vanilla while z_{context} alone provides little. We attribute these gains to the message passing mechanism of the HGNN, which aggregates information across relational edges during embedding construction; even when neighborhood nodes do not themselves encode a strong membership signal, message passing can amplify subtle parent level differences and reshape the geometry of the representation space. Across all three datasets, attacking either z_{parent} or z_{context} in isolation can yield stronger attack performance than attacking z_{final}. However, under our threat model an adversary does not have a priori knowledge of which component contains the dominant membership signal for a given dataset or generative model. MT-MIA therefore relies on z_{final} as a robust, model agnostic attack strategy that does not require such prior assumptions.

The availability of separate embedding channels provides a useful diagnostic for internal auditing and model development. By independently probing z_{parent} and z_{context}, practitioners can identify whether privacy leakage primarily arises from memorization of parent attributes or from recurring relational motifs in child tables. This decomposition enables targeted mitigation strategies, such as regularizing parent representations when leakage concentrates in z_{parent} or modifying relational modeling or sampling procedures when leakage is driven by neighborhood structure.

3.6.3 Implied Privacy of User Items

While the prior subsections audit privacy at the entity level, Theorem 4 also implies a result about item level privacy: an item's privacy is bounded by its least private representation across all connected tables in the relational schema. To empirically validate this "weakest link" effect, we compare MT-MIA against single table Distance to Closest Record (DCR) attacks applied in isolation to the child tables of each dataset for ClavaDDPM in Table 3.4. With the exception of California, single table DCR attacks detect little privacy leakage when applied to child tables. This is consistent with the i.i.d. assumption underlying these attacks, which treats each child observation as an independent sample and ignores the structural constraints imposed by the parent child join relationships. MT-MIA, by contrast, utilizes information for the entire subgraph and treats membership inference on child items as inference on the overall entity. On Airlines and California, MT-MIA substantially outperforms the single table baseline in the TPR@FPR regime, recovering leakage that the localized audit misses entirely. The implication is operationally important for relational synthetic data release: a user item that appears safe under a single table audit can leak privacy when its considered together with its neighborhood of parents and siblings.

3.6.4 Limitations

Despite the efficacy of MT-MIA, several limitations warrant further discussion.

Threat Model Constraints MT-MIA operates under a No-Box threat model in which the adversary possesses only the synthetic output G_{synth} and the database schema. As discussed in Section 3.3.4, this is the threat model that most closely matches how synthetic relational data is released in practice and is what makes MT-MIA model agnostic and dataset agnostic. The broader lesson of MT-MIA, however, is that learning an embedding of the full entity subgraph is what surfaces inter-table leakage, and this lesson should extend to less conservative threat models: a Calibrated No-Box or Shadow-Box attack that operates over learned subgraph embeddings rather than single rows would inherit the same advantage MT-MIA demonstrates over single table baselines. We see adapting these stronger attacks to the relational setting through learned representations as a natural direction for future work.

Disjoint Entity Subgraphs Theorem 4 and the auditing setup in Section 3.3.3 assume that user entities decompose into disjoint connected subgraphs, which holds for schemas where each user's data is fully separable from every other user's. Schemas with shared reference tables, such as a `Products` table referenced by every user's transactions, or schemas with many to many relationships induce overlapping

entity subgraphs in which the disjointness assumption does not hold cleanly. The MT-MIA encoder itself does not require disjointness, since the HGNN operates on whatever subgraph it is given; what changes is the auditing semantics, as a shared node in a member subgraph is necessarily also a node in some non member subgraph. Defining the appropriate unit of privacy under shared references and many to many joins is a modeling choice that depends on the auditor's goals, and we leave a systematic study of subgraph definition under these schemas to future work.

Sensitivity to Embedding Quality As a representation learning based attack, the success of MT-MIA is intrinsically tied to the discriminative power of the latent space learned by the HGNN. Graph Neural Networks are known to be sensitive to structural noise and hyperparameter configurations such as learning rate, message passing depth, and pooling strategies [71–73]. Our experiments suggest that MT-MIA remains robust across the schemas we evaluate, but this sensitivity warrants attention when applying the attack to new domains.

Signal Integration and Gated Fusion The Dynamic Gating Unit adaptively weights z_{parent} and z_{context} but does not always yield a composite embedding that is more discriminative than the individual signals. As discussed in Section 3.6.2, the raw parent or context vectors independently achieve stronger attack performance than the fused z_{final} on several configurations. Under our threat model an adversary does not know in advance which channel will dominate for a given generator and dataset, so MT-MIA defaults to z_{final} as a robust strategy, but a more principled fusion mechanism that reliably matches or exceeds the best individual channel is a direction for future work.

3.7 Conclusion

We present the first systematic study of user-level privacy auditing for synthetic relational data generation, demonstrating both theoretically and empirically that multi-table settings introduce privacy leakage at a user-level. Our proposed Multi-Table Membership Inference Attack (MT-MIA) leverages heterogeneous graph neural networks in a self-supervised manner to detect membership information leakage across connected entities without requiring generator access. Evaluation across multiple real-world datasets shows MT-MIA consistently improves upon existing single-table approaches, particularly in the critical low false-positive regime, revealing that state-of-the-art relational generators leak membership information under conservative threat models.

There are many directions for future work in this area. Efforts could focus on refining HGNN architectures

to improve the fidelity of learned subgraph representations, which would likely enhance attack performance. Extending the embedding based approach of MT-MIA to less conservative threat models would also be valuable, particularly if specific architectures become popular for relational data generation. Additionally, developing user-level differentially private relational data generators would likely be valuable in protecting user privacy. Finally, this work opens up additional lines of inquiry in extending other privacy auditing paradigms such as link prediction and attribute inference to the relational data setting.

3.8 Ethical Considerations

This work develops an attack against released synthetic relational data and demonstrates that existing generators leak membership information at the user level. We consider the ethical implications across the relevant stakeholders.

Stakeholders. The primary stakeholders are individuals whose records appear in relational databases that may be released as synthetic data, particularly in domains where membership itself is sensitive (healthcare records, financial transactions, social services interactions). Secondary stakeholders include data curators who release synthetic relational data and assume that synthesis is sufficient to protect contributors, researchers developing synthetic relational data generators, and practitioners conducting privacy audits.

Impact of the research process. MT-MIA was developed and evaluated using publicly available datasets that were released for research purposes. No additional individual data was collected, and no real synthetic data deployments were attacked.

Impact of publication. Publishing MT-MIA presents a tension. The attack could in principle be used by an adversary against a released synthetic relational dataset, particularly one generated by ClavaDDPM or RelDiff, the generators we evaluate. However, the alternative of not publishing leaves data curators with no auditing tool for user level privacy in the relational setting and no awareness that single table audits underestimate the privacy risk of multi table releases. Our judgment is that the population of curators who would benefit from understanding this risk substantially exceeds the marginal capability publication provides to adversaries, who can already attempt single table attacks. We further argue that adversarial auditing is a precondition for the development of user level differentially private relational generators (Section 3.6.4), which would address the underlying vulnerability.

Decision to publish. We considered whether to disclose this vulnerability privately to authors of the evaluated generators before publication. We decided against this for two reasons: first, the vulnerability is structural rather than implementation specific, so no patch is available for individual generators; second, the affected population (data curators considering relational synthetic data release) is broad and not tied to any

single project, making coordinated disclosure infeasible. We instead release MT-MIA as an auditing tool alongside the paper.

3.A Proofs

Theorem 1. *Let $G_{test} = (V, E)$ be a graph that is the disjoint union of two subgraphs G_{member} and $G_{holdout}$, where $V(G) = V(G_{member}) \cup V(G_{holdout})$ and $V(G_{member}) \cap V(G_{holdout}) = \emptyset$. Furthermore, there are no edges in G connecting vertices between G_{member} and $G_{holdout}$. Let $h^* \subseteq G$ be a connected subgraph, and let $g \subseteq h^*$.*

Then, if $g \subseteq G_{member}$, it follows that $h^* \subseteq G_{member}$. Likewise, if $g \subseteq G_{holdout}$, then $h^* \subseteq G_{holdout}$.

Proof. We prove the first statement; the second follows by symmetry.

Assume $g \subseteq G_{\text{member}}$ and suppose, for contradiction, that $h^* \not\subseteq G_{\text{member}}$. Then there exists at least one vertex $v \in V(h^*)$ such that $v \in V(G_{\text{holdout}})$.

Since $g \subseteq h^*$ and $g \subseteq G_{\text{member}}$, there exists at least one vertex $u \in V(g) \subseteq V(G_{\text{member}})$.

Since h^* is connected, there must exist a path P in h^* from u to v. As $u \in V(G_{\text{member}})$ and $v \in V(G_{\text{holdout}})$, this path must contain an edge (v_i, v_{i+1}) where $v_i \in V(G_{\text{member}})$ and $v_{i+1} \in V(G_{\text{holdout}})$.

However, by assumption, no edges exist in G between vertices of G_{member} and G_{holdout}. This contradiction proves that $h^* \subseteq G_{\text{member}}$.

The second statement follows by an identical argument with the roles of G_{member} and G_{holdout} reversed. □

3.B Notation

Table 3.7 summarizes the symbols introduced in Section 3.4.

3.C Experiments/ Reproducibility

3.C.1 MT-MIA Model Components

The MT-MIA architecture consists of four distinct functional stages designed to balance local record features with global relational structure:

- **Heterogeneous Message Passing (Encoder):** The core of the model utilizes two stacks of `GNNEncoder` layers, transformed via the `to_hetero` utility. Each stack employs **GATv2** (Graph

Attention Network v2) layers with a hidden dimension $d = 1024$, allowing the model to learn type-specific relationships across the relational schema.

- **Target and Context Isolation:**
  - *Target Signal (z_{target}):* Node embeddings for the target node type (the record being audited) are isolated and processed via `LayerNorm`.
  - *Context Signal ($z_{context}$):* An attention-based global pooling mechanism aggregates features from all non-target node types, representing the "relational neighborhood" of the target record.
- **Gated Fusion Mechanism:** To prevent over-reliance on either the target record or its context, a gating unit calculates a scalar $g \in [0, 1]$ via a sigmoid activation:

$$g = \sigma(W_{\text{gate}}[z_{\text{target}} \parallel z_{\text{context}}] + b_{\text{gate}}) \tag{3.7}$$

The final representation is computed as a gated residual connection:

$$z_{\text{final}} = z_{\text{target}} + (g \odot \text{Transform}(z_{\text{context}})) \tag{3.8}$$

- **Structural Anchors (Decoders):** Reconstruction heads for both target and context map hidden representations back to original feature dimensions. This acts as a structural anchor, ensuring the latent space preserves the physical characteristics of the data.

3.C.2 Hyperparameter Configuration

Table 3.5 summarizes the primary architectural parameters used in the MT-MIA implementation.

Table 3.5: MT-MIA Architectural Hyperparameters

Parameter	Value	Description
`hidden_channels`	1024	Dimensionality of the latent space.
`num_conv_stacks`	2	Number of het. message-passing blocks.
`conv_operator`	GATv2	Graph Attention Network v2 operator.
`aggregation`	Attn Pool	Weights and sums node embeddings.
`activation`	ReLU	Non-linear activation function.
`gate_activation`	Sigmoid	Actv. for the context influence gate.

3.C.3 Section 3 Experiment Details

We demonstrate how inter-table dependencies can leak membership information through a controlled experiment with synthetic relational data. This section provides complete details on the experimental setup and

Table 3.6: Dataset links, configurations and sample sizes across experimental runs.

Dataset	Schema	Training Size	Total Records
California	Households → {Individuals}	1000	3,808
Airlines	Loyalty History → Activity	1000	10,220
Airbnb	Users → Sessions	1000	24,552

methodology referenced in the main paper.

We constructed a database with two tables: Customers (parent) and Transactions (child), connected through a one-to-many relationship. Both tables contained entities with 8-dimensional feature vectors sampled from a standard multivariate Gaussian distribution $N(0, I)$. The database contained no additional attributes beyond these feature vectors and the necessary primary/foreign keys establishing relationships between tables.

For our experiment, we generated 1000 customer entities and established different relationship patterns between members and non-members. Specifically, customer entities in the training set (G_{mem}) were each associated with 100 transactions, resulting in 100,000 total transaction records. In contrast, customer entities in the test set ($G_{non\text{-}mem}$) were each associated with exactly 1 transaction, resulting in 1000 total transaction records. The synthetic data generator (G_{synth}) was trained to mimic the training set, preserving the same structural patterns and feature distributions.

The evaluation dataset was balanced with a 50:50 ratio between member and non-member records. The membership inference task involved determining whether a customer record belonged to the training data used to generate G_{synth}.

3.C.4 Section 5 Experiment

Datasets

We evaluated MT-MIA on three benchmark multi-table relational datasets with different schema structures and entity relationships, as summarized in Table 3.6.

3.C.5 Membership Inference Attack Descriptions

Distance to Closest Record (DCR. Distance-based membership inference attacks [10] are based on the intuition that synthetic data models may memorize training examples, leading to synthetic samples that lie closer in feature space to training members than to non-members. The Distance to Closest Record (DCR) method [10] formalizes this intuition by defining $f_{\text{DCR}}(x^*, S) = -\min_{x \in S} d(x^*, x)$, where $d(\cdot, \cdot)$ is a chosen

distance metric.

Density Estimation. In line with the memorization hypothesis of [10], [15] and [30] present a simple strategy of rather than computing a distance, instead estimating the density of x^* over the synthetic dataset: $f_{\text{Density Estimate}}(x^*, S) = p_S(x^*)$ using a Kernel Density Estimator.

Monte Carlo (MC). The Monte Carlo attack [45] probes overfitting by counting how often synthetic samples fall near a query. Defining the ε-neighborhood around x^* as $U_\varepsilon(x^*) = x' \mid d(x^*, x') \leq \varepsilon$, the method estimates the probability mass in this region by drawing n samples $s_1, \ldots, s_n$ from S and computing $f_{\text{MC}}(x^*, S) = \frac{1}{n} \sum_{i=1}^{n} \mathbb{I}(s_i \in U_\varepsilon(x^*))$.

Methodology

For our experiments, we defined a "user subgraph" as the complete disjoint subgraph centered around a single user entity (parent node) in the relational database, including all its connected child entities across tables. This approach allows us to sample coherent relational data structures that maintain referential integrity.

To ensure proper evaluation of membership inference, we constructed training and holdout sets by sampling entirely disjoint user subgraphs. This sampling strategy guarantees that no entity (whether parent or child) appears in both the training and holdout sets, eliminating any potential data leakage during evaluation. We only included users with at least one node in each table of the dataset's schema to ensure consistent relational structures across all sampled subgraphs.

For all experiments, we employed ClavaDDPM and RealTabFormer with default hyperparameters as implemented in the original paper. No dataset-specific modifications or hyperparameter tuning was performed, as our goal was to evaluate MT-MIA's effectiveness under standard synthetic data generation conditions rather than optimizing synthetic data quality for each specific dataset.

We applied consistent feature processing across all datasets to prepare the data for both synthetic generation and attack model training:

Feature Selection: We excluded uninformative features such as ID columns from the feature space, as these are already represented in the graph structure. Date-time variables and open text fields were also dropped due to their high dimensionality and sparsity.

Categorical Features: All categorical variables were ordinal encoded before training.

Numeric Features: All numeric features were scaled using standard scaling with parameters fit to the synthetic data and applied consistently across real and synthetic samples.

Missing Values: Missing values were ordinally encoded with 0s for categorical data or the feature mean for numeric values.

3.D Additional Tables and Figures

Table 3.7: Notation summary for Section 3.4.

Symbol	Meaning
Encoder (Sec. 3.4.1)	
$\mathcal{M}_\theta$	Heterogeneous graph encoder with parameters θ
L	Number of message passing layers
$H_l^{(t)}$	Layer l embeddings for nodes of type t
$\Phi_l^{(t)}$	Type specific message passing function at layer l
$A^{(r)}$	Adjacency matrix for relation type r
z_{parent}, z_{context}, z_{final}	Parent, relational context, and fused embeddings
g	Gating vector in $[0,1]^d$
σ	Elementwise sigmoid activation
MLP_{gate}	Multilayer perceptron producing the gate
φ	Non linear transformation in the gated residual
$\odot$	Hadamard (elementwise) product
Training (Sec. 3.4.2)	
$\mathcal{D}_\phi, \mathcal{D}_\psi$	Parent and context reconstruction decoders
ϕ, ψ	Decoder parameters
$\mathcal{N}(i)$	Neighbor set of node i across all adjacent node types
λ_p, λ_c	Reconstruction loss weights
$\mathcal{L}_{\text{recon}}$	Composite reconstruction loss
$f(h^*)$	Membership scoring function

Table 3.8: Per-seed mean and standard deviation for the comparison of MT-MIA against the best baseline attack. Values are reported as mean$_{\pm\sigma}$ over three seeds. For each (model, dataset, metric) row, *Baseline* reports the score of the strongest baseline attack on that configuration. This is the full-precision version of Table 3.1.

Model	Metric	California		Airbnb		Airlines	
		Baseline	MT-MIA	Baseline	MT-MIA	Baseline	MT-MIA
ClavaDDPM	AUC-ROC	$0.79_{\pm 0.01}$	$0.69_{\pm 0.05}$	$0.79_{\pm 0.01}$	$0.80_{\pm 0.01}$	$0.69_{\pm 0.00}$	$0.66_{\pm 0.01}$
	TPR@FPR=0	$0.00_{\pm 0.00}$	$0.07_{\pm 0.09}$	$0.00_{\pm 0.00}$	$0.01_{\pm 0.00}$	$0.07_{\pm 0.01}$	$0.17_{\pm 0.03}$
	TPR@FPR=0.001	$0.01_{\pm 0.00}$	$0.09_{\pm 0.08}$	$0.01_{\pm 0.00}$	$0.01_{\pm 0.00}$	$0.08_{\pm 0.01}$	$0.21_{\pm 0.06}$
	TPR@FPR=0.01	$0.11_{\pm 0.02}$	$0.15_{\pm 0.08}$	$0.06_{\pm 0.01}$	$0.09_{\pm 0.01}$	$0.12_{\pm 0.00}$	$0.31_{\pm 0.05}$
RelDiff	AUC-ROC	$0.67_{\pm 0.01}$	$0.64_{\pm 0.01}$	$0.57_{\pm 0.06}$	$0.62_{\pm 0.01}$	$0.51_{\pm 0.00}$	$0.49_{\pm 0.01}$
	TPR@FPR=0	$0.00_{\pm 0.00}$	$0.15_{\pm 0.03}$	$0.00_{\pm 0.00}$	$0.03_{\pm 0.02}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$
	TPR@FPR=0.001	$0.01_{\pm 0.00}$	$0.17_{\pm 0.01}$	$0.00_{\pm 0.00}$	$0.03_{\pm 0.02}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$
	TPR@FPR=0.01	$0.13_{\pm 0.01}$	$0.22_{\pm 0.01}$	$0.02_{\pm 0.01}$	$0.06_{\pm 0.01}$	$0.01_{\pm 0.00}$	$0.01_{\pm 0.00}$
RTF	AUC-ROC	$0.57_{\pm 0.01}$	$0.52_{\pm 0.02}$	$0.58_{\pm 0.00}$	$0.58_{\pm 0.00}$	$0.54_{\pm 0.00}$	$0.51_{\pm 0.01}$
	TPR@FPR=0	$0.00_{\pm 0.00}$	$0.00_{\pm 0.01}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$0.01_{\pm 0.01}$	$0.01_{\pm 0.00}$
	TPR@FPR=0.001	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$0.01_{\pm 0.00}$	$0.00_{\pm 0.00}$	$0.02_{\pm 0.00}$	$0.01_{\pm 0.01}$
	TPR@FPR=0.01	$0.02_{\pm 0.00}$	$0.02_{\pm 0.01}$	$0.02_{\pm 0.00}$	$0.02_{\pm 0.01}$	$0.05_{\pm 0.02}$	$0.03_{\pm 0.01}$

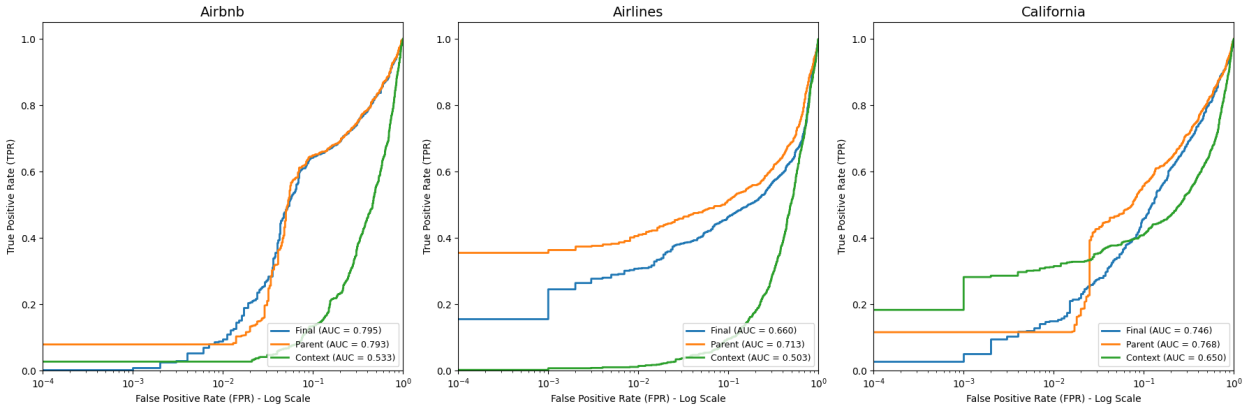


Figure 3.4: True Positive Rate by Log Scaled False Positive Rate for the most successful MT-MIA runs on ClavaDDPM. We plot the success of DCR utilizing the intermediate embeddings z_{parent} and $z_{context}$ as well as the final embedding z_{final}. While z_{final} yields a competitive attack on high fidelity multi-table data, different datasets exhibit different sources of more severe privacy leakage that are capture in these intermediate latent spaces.

Chapter 4

When Tables Leak: Attacking String Memorization in LLM-Based Tabular Data Generation

4.1 Introduction

Machine learning systems across diverse domains—from healthcare databases to financial risk assessment platforms—rely heavily on structured tabular datasets [89–91]. 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 [92]. 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 [5] and banking [6], 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[7, 8].

The contents of this chapter appeared in: Ward, J. et al., "When Tables Leak: Attacking String Memorization in LLM-Based Tabular Data Generation," Proceedings on Privacy Enhancing Technologies (PETS), 2025.

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 [93–96], 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 [4, 97, 98]. 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 [97].

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 patterns [18, 19], longer sequences [99, 100], or through supervised fine-tuning processes [101]. 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 many observations.

In order to measure privacy risk, Membership Inference Attacks (MIAs) [9, 11] 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 [10, 45, 30, 13, 77]. 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.

4.2 Related Work

To situate LevAtt within the broader landscape of synthetic data research and privacy auditing, we review prior work on tabular data generation, LLM-based modeling, and membership inference attacks, highlighting how LLMs introduce fundamentally new vulnerabilities not present in conventional generators.

4.2.1 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 [1, 2, 38], likelihood-based methods [51–53], and diffusion models [3, 75, 54] 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 of other architectures, 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 [93–96]. When larger datasets are available, SFT-based methods instead fine-tune smaller language models through direct optimization on tabular generation tasks [4, 97, 98]. Both strategies maintain the core autoregressive formulation while differing in how they leverage available data and computational resources.

4.2.2 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 [9–11]. 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:

$$\mathcal{A}(x^\star) = \mathbb{I}\left[f(x^\star) > \gamma\right] \tag{4.1}$$

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 [15, 102]. Indeed, a number of distance [10, 13], density [45, 30], and likelihood-based [14, 16, 77] 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.

4.3 Attacking Implied Strings of LLM-Based Generators

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

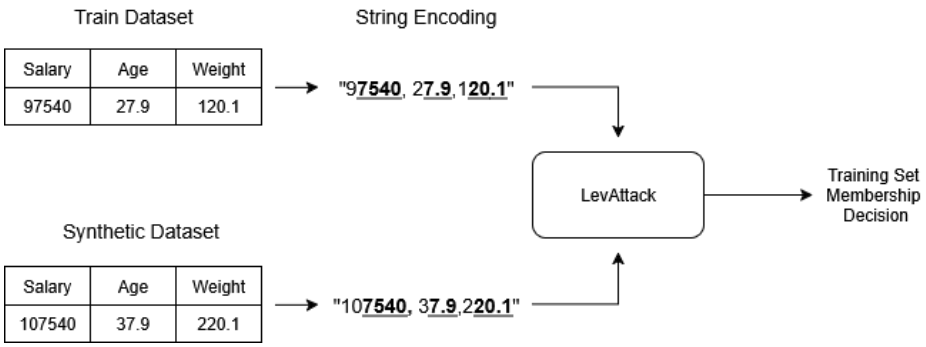


Figure 4.1: Diagram of Levenshtein Attack. We simply encode rows of tabular data into a string representation from which to attack. LevAtt finds signal in the highly constrained and often duplicated sequences of digits in synthetic tabular data generated by LLMs. In bold and underline: copied sequences of such patterns. Where these rows would be relatively far in Euclidean distance, repeated sequences in their string representations are the source of LevAtt's adversarial advantage.

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*. 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.

4.3.1 Threat Model

In this work, we adopt a conservative No-box threat model [15, 102], 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 [10, 15]. 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.

4.3.2 Levenshtein Attack

String similarity is a well-established concept, with Levenshtein edit distance providing a fundamental measure of character-level overlap [103–105]. 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 [106, 107]. However, these studies stop short of embedding string similarity into an adversarial framework, as open-ended natural language provides many valid paraphrases and thus yields weak signals for membership inference.

In contrast, *the string representations of numeric values in tabular data are highly constrained.* Unlike free-form text—where two sentences can express identical meaning with entirely different surface forms—tabular records have rigid schemas. Numerical fields follow fixed precision, delimiters appear in consistent positions, categorical fields draw from small vocabularies, and missing values are encoded in standardized ways. As a result, even a single-character modification (e.g., "17.5" → "17.6") corresponds to a meaningful change. This rigidity turns Levenshtein distance from a fragile signal in natural language into a highly sensitive indicator of memorization in tabular domains.

To operationalize this idea, we convert each record into a canonical string representation that preserves its structure. Numerical features are formatted with consistent precision, categorical variables mapped to stable tokens, and delimiters inserted to mark column boundaries. This step ensures that edit distance reflects true content similarity rather than superficial formatting differences. When a synthetic generator memorizes a training record, its outputs often contain near-exact replicas—matching digit sequences, categorical codes, and formatting patterns. Because such close matches are extremely unlikely to arise by chance, unusually low edit distances between a real record and its nearest synthetic neighbor provide strong evidence of membership.

Motivated by these observations, we design a membership inference attack based on Levenshtein distance. For each test observation x^* and synthetic set S, we extract the ordered numeric and categorical values and encode them as strings (see Figure 4.1), obtaining $x^*\text{str}$ and S_{str}. The LevAtt score for Equation 4.1 is defined as:

Table 4.1: 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)

$$f_{\text{LevAtt}}(x^*_{\text{str}}, S_{\text{str}}) = -\min_{s \in S_{\text{str}}}, \ell(x^*_{\text{str}}, s), \tag{4.2}$$

where ℓ denotes the Levenshtein edit distance. Lower distances (i.e., higher scores) indicate closer matches and therefore stronger evidence of memorization. Additionally, as 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.

4.4 Experiments

We evaluate the privacy leakage of string memorization for in-context learning (ICL) and supervised fine-tuning (SFT) 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: 4.A.

4.4.1 Experimental Design

ICL Approaches

Replicating the design of [108], we evaluate ICL approaches on the OpenML CTR23 benchmark [109], consisting of 35 real-world regression datasets 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 [110] an open-source foundation model, GPT-4o-mini

[111] a close-source foundation model, and TabPFN-v2 [112] 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 exemplar rows plus an equal holdout partition of the test set. We evaluate privacy leakage using the following No-box MIAs using the Synth-MIA package [102]: LevAtt, Euclidean Distance to Closest Record (DCR) [10], a Monte Carlo density estimation (MC) method [45], and a kernel density estimation method [15, 30]. We report MIA success using mean AUC-ROC and TPR at fixed FPR thresholds [11] to capture potential information leakage. We repeat this experimentation across 3 seeded runs.

SFT Approaches

For SFT-approaches, we benchmark the original SFT-based tabular generation method GREAT [4] and RealTabFormer [97], 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 [113]as a base model, we modify RealTabFormer to accept more modern, larger foundation models: LLaMA 3.2 (1B, 3B) [114], Qwen2.5-3B [115], and Mistral v0.3 7B [116]. Additionally, we introduce as a control CT-GAN and TVAE [38], conventional deep learning-based generators. Training follows default hyperparameters from the original GREAT and RealTabFormer implementations while CTGAN and TVAE are implemented through Synthcity [50].

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 [54], 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. Again, we repeat this experimentation across 3 seeded runs.

4.4.2 Results: ICL Methods

Our ICL experiments reveal clear evidence that LLM-based tabular generators can exhibit substantial privacy leakage, even under an extremely restrictive threat model. We organize our findings around three central observations.

LevAtt identifies substantial privacy leakage in ICL-based tabular generators. Table 4.1 reports the Mean (STD) of LevAtt AUC-ROC and TPR@FPR $\in \{0.0, 0.1\}$ across all datasets, training sizes, and seeds. Because privacy audits often emphasize worst-case behavior, we also report metrics for the top 20 runs

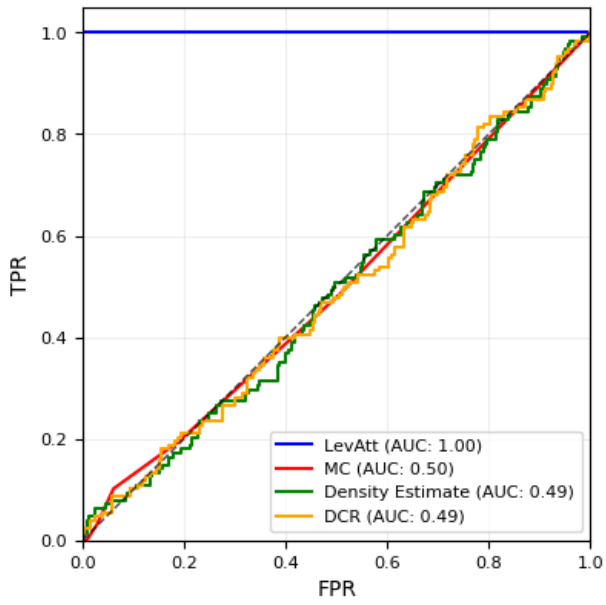


Figure 4.2: ROC plot for various No-box MIAs against TabPFN-V2 with 128 in-context samples from the MoneyBall dataset. LevAtt (blue) is able to achieve perfect classification for all in-context samples whereas MIAs that target the feature space of tabular data fail to capture the privacy leakage.

(ranked by AUC-ROC) for each model. This analysis reveals that even a simple string-similarity attack can be highly effective against state-of-the-art ICL generators. LLama 3.3-70B and TabPFN-V2 are particularly vulnerable, with mean TPR@FPR=0 values of 0.64 and 0.57 respectively in their highest-performing runs. Strikingly, LevAtt even achieves *perfect* membership classification in several TabPFN-V2 settings (Figure 4.2), demonstrating that memorized digit sequences can sometimes be reproduced verbatim in generated samples.

Privacy leakage scales with model size. These results are consistent with prior evidence that memorization grows with model capacity [18, 99]. Although the parameter counts of GPT-4o-mini and TabPFN-V2 are not publicly disclosed, empirical benchmarks place GPT-4o-mini near the 7B scale while TabPFN-V2 is lightweight enough for single-GPU inference. In contrast, the 70B-parameter LLaMA 3.3 shows the largest privacy leakage across our benchmark. This pattern reinforces that larger models, even in an ICL setting without fine-tuning, have greater capacity to memorize and regenerate specific training examples.

LevAtt captures a distinct leakage signal relative to feature-space MIAs. To evaluate whether LevAtt overlaps with traditional feature-space MIAs, we analyze its correlation with DCR, Density Estimation, and MC attacks (Figure 4.3). The three feature-space attacks are nearly perfectly correlated with each other, reflecting their shared reliance on density discrepancies. In contrast, LevAtt exhibits substantially weaker

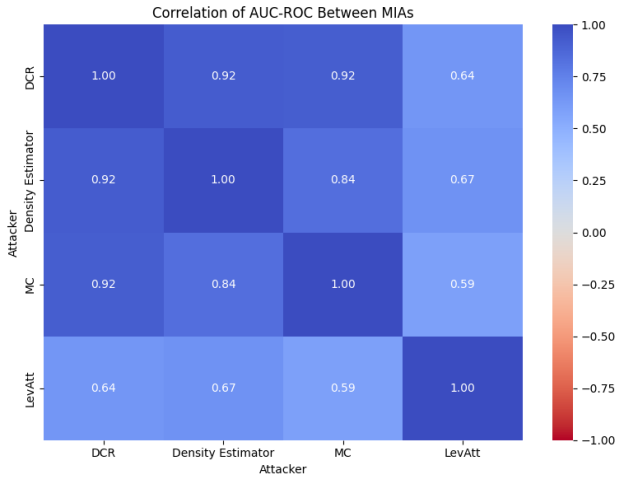


Figure 4.3: Correlation plot for No-box MIA AUC-ROC across the ICL experiment. While the feature-space targeting DCR, Density Estimate, and MC are nearly perfectly correlated, LevAtt is much less correlated. This highlights that while privacy leakage over tabular string representations and the feature space are related, LevAtt finds unique adversarial advantage.

correlation with all three, indicating that it identifies leakage signals orthogonal to feature-space methods. Indeed, in some instances, we found that LevAtt was a perfect classifier of membership in some experimental runs whereas traditional feature-space MIAs with compatible threat models were no better than random. This highlights that LLM-based tabular generators introduce a novel, string-level memorization vector that would be entirely missed by conventional synthetic-data MIAs.

4.4.3 Results: SFT Methods

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

We now turn to the SFT regime, where models are explicitly trained to approximate the full joint distribution of the training data. While leakage is generally less severe than in ICL, our findings demonstrate that SFT does not eliminate privacy risk and that leakage systematically depends on model design, sampling volume, and data structure.

SFT approaches exhibit noticeable but lower privacy leakage. Table 4.2 summarizes LevAtt performance across all datasets and seeds. Compared to ICL, SFT models typically show reduced leakage; however, RealTabFormer consistently exhibits the highest vulnerability—despite incorporating privacy-aware training—and does so more prominently than GREAT, despite their shared GPT-2 base. LevAtt also detects measurable leakage in LLaMA-3.2 (1B and 3B) and Qwen-2.5-3B. By contrast, LevAtt is ineffective against CT-GAN and TVAE, which generate full tabular rows rather than token sequences. This reinforces that

Table 4.2: 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)

token-level generation provides an attack surface absent in conventional deep-learning tabular synthesizers.

Privacy leakage increases with the volume of synthetic data. To understand how synthetic data scale affects leakage, we generate datasets at 1×, 5×, and 10× the size of the original training data using RealTabFormer. As shown in Figure 4.4, LevAtt's AUC-ROC increases monotonically with synthetic dataset size, with some datasets (e.g., Faults) exhibiting up to a 20% absolute increase. This suggests a simple mechanism: emitting more samples increases the likelihood that a memorized training example is reproduced. This has direct implications for practitioners who rely on synthetic data to replace or augment sensitive datasets—larger synthetic releases can inadvertently amplify privacy risk.

Memorization increases with sequence length in the training data. Finally, we study how digit sequence length affects memorization. Here, we create training and holdout sets using identical multivariate Gaussian distributions with a controlled numbers of digit columns. We then vary the total sequence length of digits for a row while keeping these distribution fixed. After training RealTabFormer on datasets with progressively longer sequences, we observe that LevAtt performance increases accordingly (Figure 4.5) before leveling off at 100 digits. This replicates earlier findings that longer sequences encourage LLM memorization [18] and highlights that dataset structure—not only model architecture—plays a direct role in privacy risk. Datasets containing long numeric strings (e.g., identifiers, timestamps, financial fields) inherently create more opportunities for memorization and unintended reproduction.

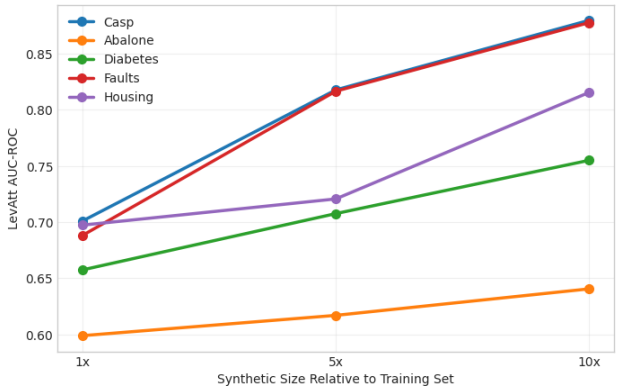


Figure 4.4: LevAtt AUC-ROC for various datasets generated by RealTabFormer with increasing synthetic dataset sizes relative to the training set.

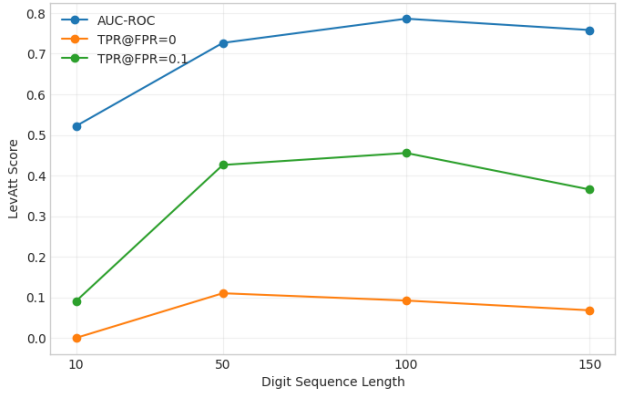


Figure 4.5: LevAtt performance on RealTabFormer at various training digit sequence lengths.

4.5 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. The first is Digit Modifier (DM), a post-processing method that operates independently of the generative model and is applied after the synthetic data have been produced. The second protection strategy is Tendency-based Logit Processor (TLP), a method compatible with any open-source LLM that strategically perturbs digits at sample time.

4.5.1 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 [117, 118], 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 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 $g : \mathbf{x_{k,i}} \times \mathbf{X_i} \rightarrow [p_{min}, p_{max}]$ 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. In our experiment, we first define

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

which represents the largest absolute value in the numerical column. We can then constructed the probability function by

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

At a high level, DM operates by iterating through each numerical entry in a tabular record, computing its digit-flipping probability using the function g, and independently perturbing each digit according to that probability. This produces a randomized yet fidelity-preserving transformation in which large-magnitude values receive stronger perturbations while small values remain close to their originals. We summarize the overall DM procedure in Algorithm 2.

Algorithm 2 Digit Modifier (DM)

Input: Training dataset $D = \{\mathbf{x}_k\}_{k=1}^{n}$, generator G, parameters $0 \leq p_{\min} < p_{\max} \leq 1$, probability function g

Output: Perturbed synthetic dataset $\tilde{D}_{\text{syn}}$

Train generator G on dataset D

Generate synthetic dataset $D_{\text{syn}} = \{\mathbf{x}_k\}_{k=1}^{n'}$ using G

Let $\mathcal{N} \subseteq \{1, \ldots, d\}$ denote the indices of numerical columns

for all $i \in \mathcal{N}$ **do**

    $\mathbf{X}_i \leftarrow \{x_{1,i}, x_{2,i}, \ldots, x_{n',i}\}$ ▷ Column i values

    $M(\mathbf{X}_i) \leftarrow \max_{\mathbf{x}_{k,i} \in \mathbf{X}_i} |\mathbf{x}_{k,i}|$ ▷ Maximum absolute value

end for

Initialize $\tilde{D}_{\text{syn}} \leftarrow D_{\text{syn}}$

for all $k \in \{1, \ldots, n'\}$ **do** ▷ For each record

    for all $i \in \mathcal{N}$ **do** ▷ For each numerical column

        $p \leftarrow g(\mathbf{x}_{k,i}, \mathbf{X}_i; p_{\min}, p_{\max})$ ▷ Compute flip probability

        Let $\mathcal{R}(\mathbf{x}_{k,i})$ denote the set of digit positions in $\mathbf{x}_{k,i}$

        for all $r \in \mathcal{R}(\mathbf{x}_{k,i})$ **do** ▷ For each digit position

            Sample $b \sim \text{Bernoulli}(p)$

            if $b = 1$ **then**

                Let d_r be the digit at position r in $\mathbf{x}_{k,i}$

                Sample $d'_r \sim \text{Uniform}(\{0, 1, \ldots, 9\} \setminus \{d_r\})$

                Replace digit at position r in $\tilde{\mathbf{x}}_{k,i}$ with d'_r

            end if

        end for

    end for

end for

return $\tilde{D}_{\text{syn}}$

4.5.2 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-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. Specifically, let $m_l = \min_j l_j$, $M_l = \max_j l_j$, and $\varepsilon > 0$ (small constant). Then, we define

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

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

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

which maps processed logits back to their original scale.

Next, TLP 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. 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

Algorithm 3 Tendency-Based Logit Processor (TLP)

Input: Training dataset D, generator G, tendency parameter $t > 0$, curving function $f_t : [0,1] \to [0,1]$
Output: Synthetic sequence generated by G with TLP applied
Train generator G on dataset D
Initialize output sequence $\mathbf{y} \leftarrow \emptyset$
while generation not complete **do**
    Compute raw logits $\mathbf{l} = (l_1, \ldots, l_k) \leftarrow G(\mathbf{y})$ ▷ k = vocabulary size
    $m_l \leftarrow \min_{j\in[k]} l_j, \quad M_l \leftarrow \max_{j\in[k]} l_j$
    $\mathbf{s} \leftarrow S_l(\mathbf{l})$ where $s_i = \frac{l_i - m_l}{M_l - m_l + \varepsilon}$ for $i \in [k]$ ▷ Scale to $[0,1]$
    $\tilde{\mathbf{s}} \leftarrow f_t(\mathbf{s})$ where $\tilde{s}_i = f_t(s_i)$ for $i \in [k]$ ▷ Apply curving function
    $\tilde{\mathbf{l}} \leftarrow S_l^{-1}(\tilde{\mathbf{s}})$ where $\tilde{l}_i = m_l + \tilde{s}_i(M_l - m_l + \varepsilon)$ for $i \in [k]$ ▷ Rescale
    $\mathbf{p} \leftarrow \text{softmax}(\tilde{\mathbf{l}})$ where $p_i = \frac{\exp(\tilde{l}_i)}{\sum_{j=1}^{k} \exp(\tilde{l}_j)}$
    Sample token $y_{\text{next}} \sim \text{Categorical}(\mathbf{p})$
    $\mathbf{y} \leftarrow \mathbf{y} \cup \{y_{\text{next}}\}$
end while
return y

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 4.6 below demonstrates the graph of our f_t under different t. Figure 4.7 shows how f_t transform the distribution of logits so that lower logits get higher probability of being selected. We summarize the overall procedure of TLP in Algorithm 3. Please refer to appendix 4.B for more details on DM and TLP.

4.5.3 Evaluating Defense Mechanisms

To evaluate the effectiveness of DM and TLP, we use RealTabFormer, the SFT model exhibiting the highest degree of privacy leakage from LevAtt, and measure how much privacy improvement each method provides against LevAtt while preserving synthetic data fidelity. Here, we use two high privacy leakage datasets for RealTabFormer: a simulated dataset from a Multivariate Gaussian N(300, 5) designed to have 100 digits, and the CASP dataset. We copy the experiment design of Section 4.4.1 and sample models at varying synthetic set sizes to induce different levels of privacy leakage in the synthetic datasets.

We then apply DM and TLP across different scaler and tendency parameter levels respectively, and assess performance in terms of privacy (LevAtt AUC-ROC and TPR@FPR=0.1), fidelity (Wasserstein Distance and Maximum Mean Discrepancy (MMD) between training and synthetic datasets), and utility (RMSE of XGBoost models trained on synthetic data and evaluated on real holdout data) [50].

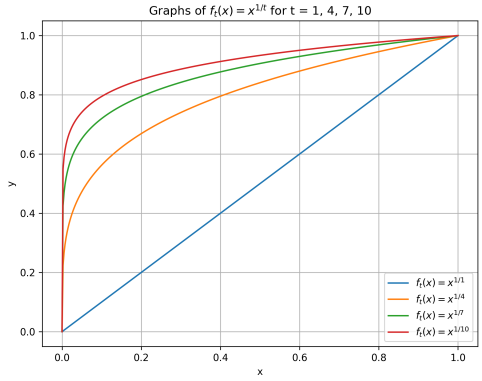


Figure 4.6: Visualization of the transformation function $f_t(x) = x^{1/t}$ under varying values of t. As t increases, the function becomes more concave, amplifying smaller logits proportionally more than larger ones. This behavior helps compress logit differences while preserving their ordering.

Overall, we find that while effective in reducing privacy leakage, DM suffers large fidelity costs. In Figure 4.8, we applied DM to the simulation dataset, which is highly vulnerable to LevAtt (showing an MIA AUC-ROC above 83% without any protection). As we increased the amount of noise injected into the synthetic data, LevAtt's AUC-ROC decreased. However, the fidelity gap measured by Wasserstein between the original training data and the noised synthetic data rose sharply. This sharp trade-off stems from the simulation dataset's small dynamic range and low variance - even small amounts of noise push values into out-of-domain regions. This illustrates that while DM is convenient- it can be applied post-hoc to any synthetic dataset- it is not an elegant protection mechanism: its agnostic approach to noise injection cannot respect the structural constraints that make synthetic data useful.

On the other hand, TLP- when equipped with a tuned tendency parameter t-controllably reduces attack efficacy while preserving the fidelity between the processed synthetic data and the training data. In our experiment, we begin with a small t and evaluate its privacy protection effect. If the resulting synthetic data does not meet the privacy threshold (LevAtt AUC–ROC below 0.55 or TPR below 0.125 at FPR $= 0.1$), we increment t and repeat the procedure, stopping once we identify the smallest t that satisfies the criterion. Across both the highly unprivate simulation setting and the CASP dataset, this tuned TLP reduces LevAtt's AUC from as high as 0.79 to 0.55 and drives the TPR@FPR=0.1 from 0.48 to below 0.125, all while incurring virtually no penalty in the Maximum Mean Discrepancy of the TLP-generated data across all synthetic data sizes (see Figure 4.9).

TLP not only preserves the fidelity of the training dataset while meeting the privacy threshold, but also maintains the downstream utility of the CASP dataset. To evaluate this, we train XGBoost models on

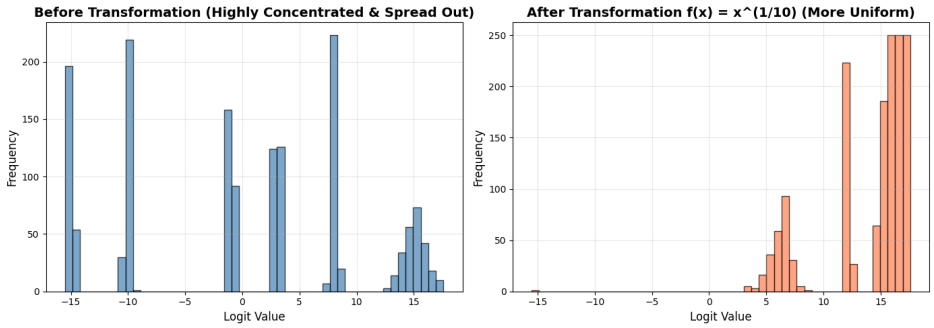


Figure 4.7: Effect of the TLP transformation on logit distributions. Before transformation (left), lower logits are tightly concentrated near the bottom of the range and have little chance of being selected. After applying the TLP function $f(x) = x^{1/10}$ (right), the transformation increases their relative magnitude of lower value logits, making them more likely to be sampled.

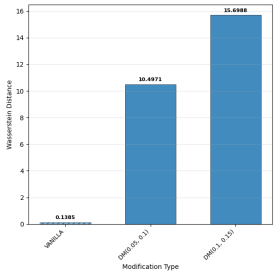
three versions of the data: real training subsets, vanilla synthetic data generated without protection, and TLP-processed synthetic data that satisfies the privacy requirement (LevAtt AUC-ROC below 0.55 or TPR below 0.125 at FPR = 0.1). For each training size, model performance is assessed on the same held-out real test set using RMSE, and we observe that the utility degradation remains within a reasonable range (see Figure 4.10).

4.6 Discussion

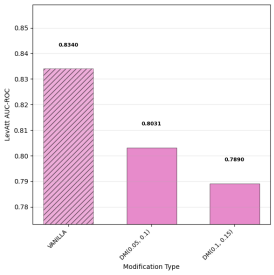
To contextualize our findings and their implications, we discuss LevAtt's broader significance, evaluate the effectiveness of proposed defenses, examine what LLMs truly learn from tabular data, and outline limitations and future research directions.

4.6.1 LevAtt and the Privacy of LLM-Based Synthetic Data Generation

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 uncovers that state-of-the-art ICL models can catastrophically leak training membership by copying sequential patterns of digits and text from prompt exemplar 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. In



(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.

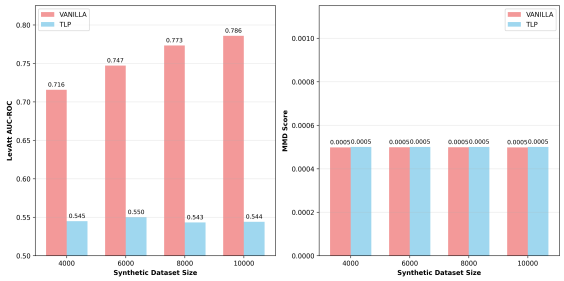
Figure 4.8: Privacy–fidelity comparison of DM on RealTabFormer synthetic data from the simulated dataset (Section 4.5.3). Vanilla corresponds to plain sampling without protection. Panel (a) reports Wasserstein distances; panel (b) shows LevAtt AUC-ROC. While DM is able to induce reductions in privacy leakage, the resulting synthetic data are of low fidelity.

contrast, conventional tabular generators such as CT-GAN and TVAE were resistant to string-based attacks, and LevAtt exhibited low correlation with feature-space oriented MIAs.

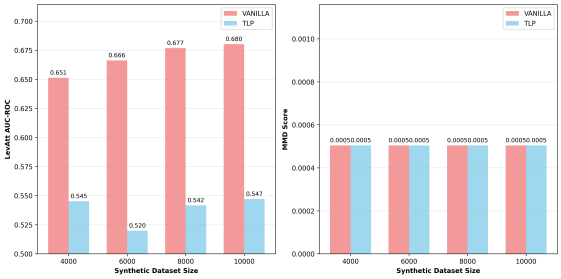
This vulnerability stems from fundamental differences in how LLMs generate synthetic data. Unlike GANs and VAEs that model joint distributions directly in the feature space, LLMs decompose generation into sequential token prediction, where each digit is conditioned on previously generated values. This autoregressive process creates opportunities for the model to reproduce memorized digit sequences, particularly when training data contains repeated patterns, long numeric strings, or low-variance columns—structural characteristics common in real-world tabular datasets. These results highlight that autoregressive token-based generation in LLMs exposes a distinct attack surface not shared by other deep learning architectures, emphasizing the need for dedicated privacy audits in this domain.

4.6.2 Defenses for LLMs

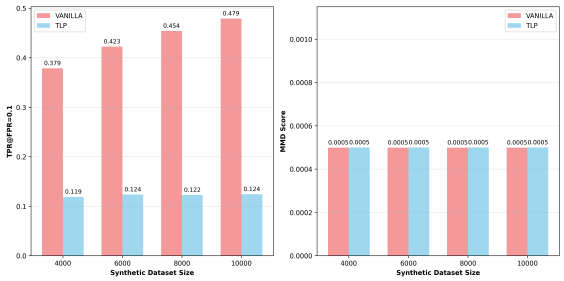
While highly deployable and powerful, LevAtt can be defeated. In this work, we proposed an intuitive post-hoc defense called Digit Modifier (DM), which alters digits after generation. However, we found that DM failed to preserve the fidelity of the synthetic dataset. In contrast, our Tendency-based Logit Processor (TLP) successfully reduced LevAtt's AUC-ROC to below 55% and TPR to below 12.5% at FPR=10% across both simulation and real-world datasets, while maintaining Maximum Mean Discrepancy nearly identical to vanilla generation. DM, despite achieving similar privacy improvements, incurred substantial fidelity costs,



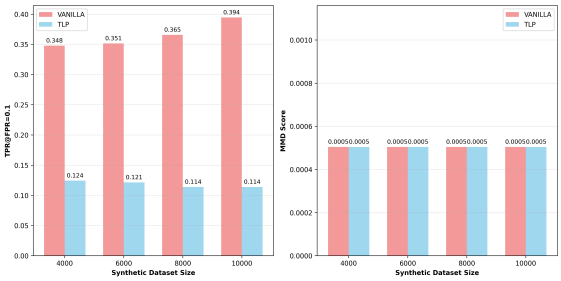
(a) Simulation Dataset- LevAtt AUC and MMD



(b) CASP Dataset- LevAtt AUC and MMD



(c) Simulation Dataset- LevAtt TPR@FPR=0.1 and MMD



(d) CASP Dataset- LevAtt TPR@FPR=0.1 and MMD

Figure 4.9: Privacy–fidelity trade-off of TLP on RealTabFormer synthetic data. We plot the AUC and Maximum Mean Discrepancy (MMD) for Vanilla and TLP-protected RealTabFormer models in Panels (a)-(b). For both simulation (Section 4.5.3) and CASP datasets, TLP consistently reduces privacy leakage (AUC-ROC to ∼55%) while creating synthetic data that matches the MMD of unmodified Vanilla data at various sizes. This trend holds as well for TPR@FPR=0.1 in Panels (c)-(d) where TLP successfully reduces the TPR@FPR=0.1 to below 12.5% without loss to fidelity.

with Wasserstein distances increasing sharply as noise injection intensified.

The key advantage of TLP lies in its integration with the generation process itself. By perturbing logits during inference, TLP allows the model to maintain coherent dependencies across features while strategically introducing uncertainty in high-confidence digit predictions. DM, operating post-hoc, must blindly flip digits without access to the model's learned correlations, making it difficult to balance privacy protection with fidelity preservation. This is particularly problematic for datasets with narrow distributions or tight inter-feature dependencies, where even minor perturbations can push synthetic samples into low-density regions. TLP can be appended to any open-source LLM through the Hugging Face API, effectively controlling privacy leakage by smoothing the logits of digits with disproportionately high probabilities without substantially altering the statistical fidelity of the resulting synthetic data.

4.6.3 On the Nature of LLM-Based Tabular Learning

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.

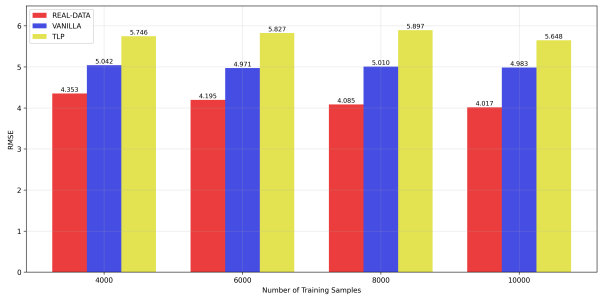
This question is not merely theoretical—our experimental results provide concrete evidence of memorization-like behavior. The perfect membership classification achieved by LevAtt on some TabPFN-V2 runs, combined with the scaling of privacy leakage with model size, suggests that larger models increasingly rely on storing and retrieving training examples rather than abstracting general patterns. Furthermore, our finding that privacy leakage increases with both the volume of synthetic data generated (Figure 4.4) and the length of digit sequences in training data (Figure 4.5) aligns more closely with a retrieval mechanism than with principled distribution learning.

This question echoes similar debates in natural language processing, where LLMs demonstrably memorize training data under certain conditions [18]. However, tabular data may be particularly susceptible to memorization due to its rigid structure and lack of paraphrase. In natural language, identical semantic content can be expressed in countless surface forms, creating ambiguity about whether a model has memorized a specific sentence or learned underlying concepts. Tabular data offers no such ambiguity—a sequence of digits either matches or does not. This rigidity may push LLM-based tabular generators toward memorization even when natural language applications achieve genuine generalization. Understanding when LLMs genuinely learn tabular distributions versus when they rely on approximate memorization remains an important direction for future work.

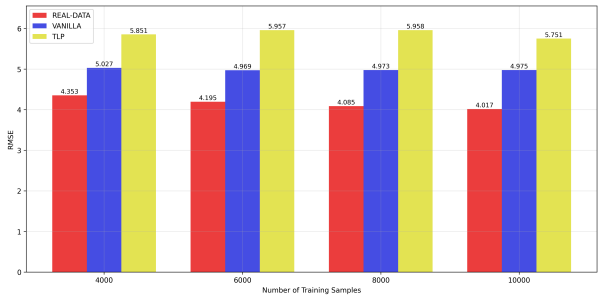
4.6.4 Limitations

While LevAtt and our proposed defenses demonstrate significant privacy vulnerabilities and mitigation strategies, several limitations warrant discussion.

First, LevAtt operates under a conservative No-box threat model, which—while demonstrating that privacy leakage occurs even under minimal adversarial assumptions—likely underestimates the true privacy risk in other possible scenarios. More permissive threat models that grant adversaries knowledge of model implementation details, training hyperparameters, or access to reference datasets would enable substantially



(a) RMSE of CASP dataset XGBoost models where TLP-protected datasets have a LevAtt AUC of below 0.55.



(b) RMSE of CASP dataset XGBoost models where TLP-protected datasets have a LevAtt TPR@FPR=0.1 of below 0.125.

Figure 4.10: Utility comparison of XGBoost models trained on real, vanilla synthetic, and TLP-protected synthetic data at various synthetic dataset (and thus privacy leakage) sizes. Real data achieves the lowest RMSE, vanilla synthetic shows moderate degradation, and TLP-protected data shows larger degradation. However, the gap between vanilla and TLP remains stable as training size increases, indicating that TLP provides a controllable privacy–utility trade-off even when stronger tendency levels are needed.

more powerful attacks. For instance, knowledge of the specific tokenization scheme or access to auxiliary data could allow adversaries to adapt existing LLM-based membership inference techniques from natural language to tabular domains, potentially achieving even higher attack success rates than those reported here.

Second, while DM and TLP effectively reduce LevAtt's success rate, neither method provides formal privacy guarantees comparable to differential privacy. Both defenses are empirically validated against a specific attack rather than offering provable protection against arbitrary privacy audits. Consequently, more sophisticated attacks—particularly those that exploit model properties beyond string similarity—may successfully defeat these defenses [119–121].

Third, we do not address adaptive adversaries who are aware that protective mechanisms are in place. An adversary with knowledge of TLP deployment, for example, might develop attacks that specifically target the statistical artifacts introduced by logit perturbation, or exploit correlations that remain intact despite digit modifications. Such adaptive strategies could potentially circumvent our defenses, highlighting the need for more robust, defense-aware attack evaluations and iterative improvement of protection mechanisms.

4.6.5 Future Work

Our findings open several avenues for future research aimed at both strengthening the understanding of privacy leakage in LLM-based tabular data generators and developing more robust protective mechanisms.

A first direction is the exploration of richer threat models beyond the No-box setting. While our

results show that even minimally informed adversaries can perform highly accurate membership inference, more permissive models may reveal additional vulnerabilities. Future work could investigate attacks that leverage model internals, surrogate data, or tokenization details, and evaluate how such enhanced adversarial capabilities affect memorization dynamics in structured data domains. This includes adapting or extending gradient-based or embedding-space attacks from NLP to tabular contexts, potentially uncovering deeper patterns of leakage.

Second, our proposed defenses—while effective against LevAtt—lack provable privacy guarantees. Future research should aim to establish theoretical foundations for privacy in LLM-based tabular generation, analogous to differential privacy frameworks used in classical synthetic data generation. One promising direction is designing logit- or representation-level perturbation mechanisms that preserve distributional fidelity while offering quantifiable bounds on memorization risk. Similarly, exploring how architectural choices, training objectives, or regularization schemes influence memorization could inform principled defense design.

4.7 Conclusion

In this work 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 learn and represent tabular distributions.

4.8 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.

4.9 Acknowledgments

The authors 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 revisions by LLMs were checked and validated by the authors.

4.A Appendix

4.A.1 In-Context Learning details

Generator Details

1. **TabPFN Generation [112]**: 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.

2. **LLaMA Generation [110]**: 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.[1]

[1]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.

3. **GPT-4o-mini [111]**: 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.

Listing 4.1: ICL Prompt Template 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}

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.
```

4.A.2 Black-Box MIAs

We apply feature space based black box MIAs in the same experimental design as LevAtt. Here, we treat the exemplar observations as the membership class and holdout data as non-membership. Before calling the attacks, we scale continuous and one-hot encode categorical variables for DCR and MC. As Kernel Density Estimation often fails to converge under one-hot encoded categorical variables we instead ordinal encode them. All implementations of these attacks are through the Synth-MIA library [102].

1. **Distance to Closest Record (DCR)** [10]: DCR is a black-box attack that scores test data based on a score of the Euclidean distance to the nearest neighbor in the synthetic dataset.
2. **MC** [45]: MC is based on counting the number of observations in the synthetic dataset that fall into the neighborhood of a test point (Monte Carlo Integration). However, this method does not consider a reference dataset, and the choice of distance metric for defining a neighborhood is a non-trivial hyperparameter to tune.
3. **Density Estimate** [45, 30]: Density Estimate follows a similar strategy as MC, but rather than using a Monte Carlo approximation of density, instead uses a Kernel Density Estimator (KDE). The idea is that a test observation that is a training observation will have a higher density estimate on a KDE fit over a synthetic dataset.

4.A.3 SFT-Datasets

Dataset	# Instances	# Features
Abalone (OpenML)	4,177	9
CA Housing (OpenML)	20,640	9
CASP (OpenML)	45,730	9
Diabetes (Sklearn)	412	10
Steel Plates Faults (UCI)	1,941	27

4.A.4 SFT Model Details

We modify the original RealTabFormer implementation to use LLaMA 3.2 (1B, 3B) [114], Qwen2.5-3B [115], and Mistral v0.3 7B [116]. Here, we follow RealTabFormer's base training and sampling hyperparameters in

Table 4.3. To SFT GREAT, we also use its original implementation of which the base hyperparameters can be found in Table 4.4. For CT-GAN and TVAE, we use the default hyperparameters and implementation found in Synthcity [50].

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 4.3: 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 4.4: GReaT training and sampling hyperparameters.

4.A.5 Simulated Data Generation Details

For Figure 4.5, 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.

4.B More Details on Defenses

Digit Substitution Rule in DM: For the Digit Modifier (DM), each selected digit is perturbed using a simple increment rule. Specifically, a digit $d \in \{0, 1, \ldots, 9\}$ is replaced by $(d + 1) \bmod 10$. Thus, 0 becomes 1, 1 becomes 2, and so on, with 9 wrapping around to 0. This cyclic increment operation provides a minimal yet consistent perturbation to each affected digit, ensuring that the modified values remain close to their originals while still introducing controlled randomness.

Handling Invalid Logits in ReaLTabFormer: In ReaLTabFormer, some logits take the value $-\infty$ or become `NaN` due to masking constraints in the tabular structure. These values arise when certain tokens are structurally disallowed, leading to zero-probability entries after masking. Since such logits cannot be meaningfully transformed by TLP, we simply ignore them: TLP is applied only to finite-valued logits, while any $-\infty$ or `NaN` logits are left unchanged to preserve the validity of the generator's masking logic.

Why TLP Is Applied Only at Inference Time: The Tendency-based Logit Processor (TLP) is designed exclusively for inference-time perturbations. Applying TLP during training severely disrupts the optimization dynamics: the modified logits distort the gradient signal, causing the loss to decrease extremely slowly and preventing the model from converging in a reasonable number of steps. In effect, the model must simultaneously learn both the underlying data distribution and the perturbation induced by TLP, which dramatically complicates training. By restricting TLP to inference time, we preserve stable training behavior while still introducing controlled variability into the generated samples.

Role of the Stabilization Constant ε. The shifted min–max scaling used in TLP requires a small positive constant ε to ensure numerical stability. In cases where all logits in a column share the same value, we have $M_l = m_l$, causing the denominator $M_l - m_l$ in the scaling expression to become zero. Without a stabilizing term, this would result in undefined values or `NaN` outputs after normalization. By adding a small $\varepsilon > 0$, we guarantee that the denominator remains strictly positive, preventing division-by-zero errors and ensuring that the scaled logits remain well defined. This safeguard is essential for applying TLP robustly, especially when the generator outputs nearly constant logits due to structural constraints in the data.

Bibliography

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

[2] 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.

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

[4] Vadim Borisov, Kathrin Seßler, Tobias Leemann, Martin Pawelczyk, and Gjergji Kasneci. Language models are realistic tabular data generators, 2023.

[5] 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.

[6] 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.

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

[8] 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.

[9] 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)*, pages 3–18, Los Alamitos, CA, USA, may 2017. IEEE Computer Society.

[10] 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. ACM, October 2020.

[11] Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, A. Terzis, and Florian Tramèr. Membership inference attacks from first principles. *2022 IEEE Symposium on Security and Privacy (SP)*, pages 1897–1914, 2021.

[12] Georgi Ganev and Emiliano De Cristofaro. On the inadequacy of similarity-based privacy metrics: Reconstruction attacks against "truly anonymous synthetic data", 2023.

[13] Joshua Ward, Chi-Hua Wang, and Guang Cheng. Data plagiarism index: Characterizing the privacy risk of data-copying in tabular generative models. *KDD- Generative AI Evaluation Workshop*, 2024.

[14] Theresa Stadler, Bristena Oprisanu, and Carmela Troncoso. Synthetic data – anonymisation groundhog day. In *31st USENIX Security Symposium (USENIX Security 22)*, pages 1451–1468, Boston, MA, August 2022. USENIX Association.

[15] 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. *arXiv preprint arXiv:2211.06550*, 2022.

[16] Matthieu Meeus, Florent Guepin, Ana-Maria Creţu, and Yves-Alexandre de Montjoye. *Achilles' Heels: Vulnerable Record Identification in Synthetic Data Publishing*, page 380–399. Springer Nature Switzerland, 2024.

[17] Steven Golob, Sikha Pentyala, Anuar Maratkhan, and Martine De Cock. Privacy vulnerabilities in marginals-based synthetic data, 2024.

[18] 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, 2021.

[19] Nikhil Kandpal, Eric Wallace, and Colin Raffel. Deduplicating training data mitigates privacy risks in language models. *ArXiv*, abs/2202.06539, 2022.

[20] Martin Abadi, Andy Chu, Ian Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, CCS'16. ACM, October 2016.

[21] Puyu Wang, Yunwen Lei, Yiming Ying, and Hai Zhang. Differentially private sgd with non-smooth losses. *Applied and Computational Harmonic Analysis*, 56:306–336, 2022.

[22] 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, August 2020.

[23] Moritz Hardt, Guy N. Rothblum, and Rocco A. Servedio. Private data release via learning thresholds. In *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '12, page 168–187, USA, 2012. Society for Industrial and Applied Mathematics.

[24] Anupam Gupta, Aaron Roth, and Jonathan Ullman. Iterative constructions and private data release. In Ronald Cramer, editor, *Theory of Cryptography*, pages 339–356, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[25] S. Takagi, T. Takahashi, Y. Cao, and M. Yoshikawa. P3gm: Private high-dimensional data release via privacy preserving phased generative model. In *2021 IEEE 37th International Conference on Data Engineering (ICDE)*, pages 169–180, Los Alamitos, CA, USA, apr 2021. IEEE Computer Society.

[26] Zilong Zhao, Aditya Kunar, Robert Birke, and Lydia Y. Chen. Ctab-gan: Effective table data synthesizing. In Vineeth N. Balasubramanian and Ivor Tsang, editors, *Proceedings of The 13th Asian Conference on Machine Learning*, volume 157 of *Proceedings of Machine Learning Research*, pages 97–112. PMLR, 17–19 Nov 2021.

[27] Morgan Guillaudeux, Olivia Rousseau, Julien Petot, Zineb Bennis, Charles-Axel Dein, Thomas Goronflot, Matilde Karakachoff, Sophie Limou, Nicolas Vince, Matthieu Wargny, and Pierre-Antoine Gourraud. Patient-centric synthetic data generation, no reason to risk re-identification in the analysis of biomedical pseudonymised data. 05 2022.

[28] Tongyu Liu, Ju Fan, Guoliang Li, Nan Tang, and Xiaoyong Du. Tabular data synthesis with generative adversarial networks: design space and optimizations. *The VLDB Journal*, 33(2):255–280, aug 2023.

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

[30] Boris van Breugel, Hao Sun, Zhaozhi Qian, and Mihaela van der Schaar. Membership inference attacks against synthetic data through overfitting detection. In Francisco Ruiz, Jennifer Dy, and Jan-Willem van de Meent, editors, *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of Machine Learning Research*, pages 3493–3514. PMLR, 25–27 Apr 2023.

[31] Jiayuan Ye, Aadyaa Maddi, Sasi Kumar Murakonda, Vincent Bindschaedler, and Reza Shokri. Enhanced membership inference attacks against machine learning models. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, CCS '22, page 3093–3106, New York, NY, USA, 2022. Association for Computing Machinery.

[32] Alexandre Sablayrolles, Matthijs Douze, Cordelia Schmid, Yann Ollivier, and Hervé Jégou. White-box vs black-box: Bayes optimal strategies for membership inference. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 5558–5567, Long Beach, CA, USA, June 2019. PMLR.

[33] Noseong Park, Mahmoud Mohammadi, Kshitij Gorde, Sushil Jajodia, Hongkyu Park, and Youngmin Kim. Data synthesis based on generative adversarial networks. *Proc. VLDB Endow.*, 11(10):1071–1083, June 2018.

[34] Pei-Hsuan Lu, Pang-Chieh Wang, and Chia-Mu Yu. Empirical evaluation on synthetic data generation with generative adversarial network. In *Proceedings of the 9th International Conference on Web Intelligence, Mining and Semantics*, WIMS2019, New York, NY, USA, 2019. Association for Computing Machinery.

[35] Andrew Yale, Saloni Dash, Ritik Dutta, Isabelle Guyon, Adrien Pavao, and Kristin P. Bennett. Assessing privacy and quality of synthetic health data. In *Proceedings of the Conference on Artificial Intelligence for Data Discovery and Reuse*, AIDR '19, New York, NY, USA, 2019. Association for Computing Machinery.

[36] Mostly AI. Truly anonymous synthetic data – evolving legal definitions and technologies (part ii), 2020.

[37] Mostly AI. How to implement data privacy? a conversation with klaudius kalcher, 2021.

[38] Lei Xu, Maria Skoularidou, Alfredo Cuesta-Infante, and Kalyan Veeramachaneni. Modeling tabular data using conditional gan. In *Neural Information Processing Systems*, 2019.

[39] Liwei Song and Prateek Mittal. Systematic evaluation of privacy risks of machine learning models. In *USENIX Security Symposium*, 2020.

[40] S. Yeom, I. Giacomelli, M. Fredrikson, and S. Jha. Privacy risk in machine learning: Analyzing the connection to overfitting. In *2018 IEEE 31st Computer Security Foundations Symposium (CSF)*, pages 268–282, Los Alamitos, CA, USA, jul 2018. IEEE Computer Society.

[41] Yunhui Long, Lei Wang, Diyue Bu, Vincent Bindschaedler, Xiaofeng Wang, Haixu Tang, Carl A. Gunter, and Kai Chen. A pragmatic approach to membership inferences on machine learning models. In *2020 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 521–534, 2020.

[42] Lauren Watson, Chuan Guo, Graham Cormode, and Alexandre Sablayrolles. On the importance of difficulty calibration in membership inference attacks. In *International Conference on Learning Representations*, 2022.

[43] Sajjad Zarifzadeh, Philippe Liu, and Reza Shokri. Low-cost high-power membership inference attacks, 2024.

[44] Jamie Hayes, Luca Melis, George Danezis, and Emiliano De Cristofaro. Logan: Membership inference attacks against generative models. *Proceedings on Privacy Enhancing Technologies*, 2019:133 – 152, 2017.

[45] 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.

[46] Frank R. Hampel. The influence curve and its role in robust estimation. *Journal of the American Statistical Association*, 69(346):383–393, 1974.

[47] R.D. Cook and S. Weisberg. *Residuals and Influence in Regression*. Monographs on statistics and applied probability. Chapman and Hall, 1986.

[48] Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine*

Learning, volume 70 of *Proceedings of Machine Learning Research*, pages 1885–1894. PMLR, 06–11 Aug 2017.

[49] Bernd Bischl, Giuseppe Casalicchio, Matthias Feurer, Frank Hutter, Michel Lang, Rafael G. Mantovani, Jan N. van Rijn, and Joaquin Vanschoren. Openml benchmarking suites. *arXiv:1708.03731v2 [stat.ML]*, 2019.

[50] Zhaozhi Qian, Bogdan-Constantin Cebere, and Mihaela van der Schaar. Synthcity: facilitating innovative use cases of synthetic data in different data modalities, 2023.

[51] Ankur Ankan and Abinash Panda. pgmpy: Probabilistic graphical models using python. In *Proceedings of the Python in Science Conference*, SciPy. SciPy, 2015.

[52] Conor Durkan, Artur Bekasov, Iain Murray, and George Papamakarios. *Neural spline flows*. Curran Associates Inc., Red Hook, NY, USA, 2019.

[53] 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, editors, *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of Machine Learning Research*, pages 5357–5375. PMLR, 25–27 Apr 2023.

[54] 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*, 2024.

[55] Joshua Ward, Bochao Gu, Chi-Hua Wang, and Guang Cheng. When tables leak: Attacking string memorization in llm-based tabular data generation, 2025.

[56] Nicola De Cao, Ivan Titov, and Wilker Aziz. Block neural autoregressive flow. *35th Conference on Uncertainty in Artificial Intelligence (UAI19)*, 2019.

[57] Hongwei Wen and Hanyuan Hang. Random forest density estimation. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 23701–23722. PMLR, 17–23 Jul 2022.

[58] Meenatchi Sundaram Muthu Selva Annamalai, Georgi Ganev, and Emiliano De Cristofaro. "what do you want from theory alone?" experimenting with tight auditing of differentially private synthetic data generation, 2024.

[59] Masashi Sugiyama, Taiji Suzuki, and Takafumi Kanamori. *Density Ratio Estimation in Machine Learning*. Cambridge University Press, Cambridge, 2012.

[60] Takafumi Kanamori, Shohei Hido, and Masashi Sugiyama. A least-squares approach to direct importance estimation. *Journal of Machine Learning Research*, 10:1391–1445, 2009.

[61] Usama Fayyad, Gregory Piatetsky-Shapiro, and Padhraic Smyth. From data mining to knowledge discovery in databases. *AI Magazine*, 17(3):37, Mar. 1996.

[62] Carmen Martínez-Cruz, Ignacio J. Blanco, and María Amparo Vila. Ontologies versus relational databases: are they so different? a comparison. *Artificial Intelligence Review*, 38:271–290, 2012.

[63] Matthias Fey, Weihua Hu, Kexin Huang, Jan Eric Lenssen, Rishabh Ranjan, Joshua Robinson, Rex Ying, Jiaxuan You, and Jure Leskovec. Position: Relational deep learning - graph representation learning on relational databases. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp, editors, *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 13592–13607. PMLR, 21–27 Jul 2024.

[64] Neha Patki, Roy Wedge, and Kalyan Veeramachaneni. The synthetic data vault. In *2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, pages 399–410, 2016.

[65] Inkit Padhi, Yair Schiff, Igor Melnyk, Mattia Rigotti, Youssef Mroueh, Pierre Dognin, Jerret Ross, Ravi Nair, and Erik Altman. Tabular transformers for modeling multivariate time series. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3565–3569, 2021.

[66] Mohamed Gueye, Yazid Attabi, and Maxime Dumas. Row conditional-tgan for generating synthetic relational databases. pages 1–5, 06 2023.

[67] Wei Pang, Masoumeh Shafieinejad, Lucy Liu, Stephanie Hazlewood, and Xi He. Clavaddpm: multi-relational data synthesis with cluster-guided diffusion models. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, NIPS '24, Red Hook, NY, USA, 2024. Curran Associates Inc.

[68] Valter Hudovernik, Minkai Xu, Juntong Shi, Lovro Šubelj, Stefano Ermon, Erik Štrumbelj, and Jure Leskovec. Reldiff: Relational data generative modeling with graph-based diffusion models, 2025.

[69] Chuxu Zhang, Dongjin Song, Chao Huang, Ananthram Swami, and Nitesh V. Chawla. Heterogeneous graph neural network. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19, page 793–803, New York, NY, USA, 2019. Association for Computing Machinery.

[70] Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Yanfang Ye, Peng Cui, and Philip S Yu. Heterogeneous graph attention network. In *The World Wide Web Conference*, WWW '19, page 2022–2032, New York, NY, USA, 2019. Association for Computing Machinery.

[71] Xinyu Fu, Jiani Zhang, Ziqiao Meng, and Irwin King. Magnn: Metapath aggregated graph neural network for heterogeneous graph embedding. WWW '20, page 2331–2341, New York, NY, USA, 2020. Association for Computing Machinery.

[72] Matthias Fey, Weihua Hu, Kexin Huang, Jan Eric Lenssen, Rishabh Ranjan, Joshua Robinson, Rex Ying, Jiaxuan You, and Jure Leskovec. Relational deep learning: Graph representation learning on relational databases, 2023.

[73] Xiaocheng Yang, Mingyu Yan, Shirui Pan, Xiaochun Ye, and Dongrui Fan. Simple and efficient heterogeneous graph neural network, 2023.

[74] Joshua Robinson, Rishabh Ranjan, Weihua Hu, Kexin Huang, Jiaqi Han, Alejandro Dobles, Matthias Fey, Jan Eric Lenssen, Yiwen Yuan, Zecheng Zhang, et al. Relbench: A benchmark for deep learning on relational databases. *Advances in Neural Information Processing Systems*, 37:21330–21341, 2024.

[75] Namjoon Suh, Xiaofeng Lin, Din-Yin Hsieh, Mehrdad Honarkhah, and Guang Cheng. Autodiff: combining auto-encoder and diffusion model for tabular data synthesizing. In *NeurIPS 2023 Workshop on Synthetic Data Generation with Generative AI*, 2023.

[76] Sajjad Zarifzadeh, Philippe Liu, and Reza Shokri. Low-cost high-power membership inference attacks, 2024.

[77] Joshua Ward, Chi-Hua Wang, and Guang Cheng. Privacy auditing synthetic data release through local likelihood attacks, 2025.

[78] Masoumeh Shafieinejad, Xi He, Mahshid Alinoori, John Jewell, Sana Ayromlou, Wei Pang, Veronica Chatrath, Garui Sharma, and Deval Pandya. Midst challenge at satml 2025: Membership inference over diffusion-models-based synthetic tabular data, 2026.

[79] Xiaoyu Wu, Yifei Pang, Terrance Liu, and Steven Wu. Winning the midst challenge: New membership inference attacks on diffusion models for tabular data synthesis, 2025.

[80] Michael Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. Modeling relational data with graph convolutional networks. In Aldo Gangemi, Roberto Navigli, Maria-Esther Vidal, Pascal Hitzler, Raphaël Troncy, Laura Hollink, Anna Tordai, and Mehwish Alam, editors, *The Semantic Web*, pages 593–607, Cham, 2018. Springer International Publishing.

[81] Ziniu Hu, Yuxiao Dong, Kuansan Wang, and Yizhou Sun. Heterogeneous graph transformer. In *Proceedings of The Web Conference 2020*, WWW '20, page 2704–2710, New York, NY, USA, 2020. Association for Computing Machinery.

[82] Shaked Brody, Uri Alon, and Eran Yahav. How attentive are graph attention networks? In *International Conference on Learning Representations*, 2022.

[83] Minnesota Population Center. Integrated public use microdata series, international: Version 7.3, 2020.

[84] Agung Pambudi. Airline loyalty campaign program impact on flights. Kaggle Dataset, 2025. Retrieved from https://www.kaggle.com/datasets/agungpambudi/airline-loyalty-campaign-program-impact-on-flights.

[85] Airbnb (via Kaggle). Airbnb recruiting: New user bookings. Kaggle Competition Dataset, 2015. Retrieved from https://www.kaggle.com/competitions/airbnb-recruiting-new-user-bookings/data.

[86] Florent Guépin, Nataša Krčo, Matthieu Meeus, and Yves-Alexandre de Montjoye. Lost in the averages: A new specific setup to evaluate membership inference attacks against machine learning models, 2024.

[87] Joshua Ward, Xiaofeng Lin, , Chi-Hua Wang, and Guang Cheng. Synth-mia: A testbed for auditing privacy leakage in tabular data synthesis, 2025.

[88] Meenatchi Sundaram Muthu Selva Annamalai, Georgi Ganev, and Emiliano De Cristofaro. "what do you want from theory alone?" experimenting with tight auditing of differentially private synthetic data generation. In *USENIX Security Symposium*, 2024.

[89] Mauro Giuffré and Dennis L. Shung. Harnessing the power of synthetic data in healthcare: innovation, application, and privacy. *NPJ Digital Medicine*, 6, 2023.

[90] 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.

[91] 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.

[92] Joao Fonseca and Fernando Bação. Tabular and latent space synthetic data generation: a literature review. *Journal of Big Data*, 10, 07 2023.

[93] 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*, volume 235, pages 44060–44092, Vienna, Austria, 2024. PMLR.

[94] 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*, Vancouver, Canada, 2024. Curran Associates, Inc.

[95] 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 2025.

[96] Junwei Ma, Apoorv Dankar, George Stein, Guangwei Yu, and Anthony Caterini. Tabpfgen – tabular data generation with tabpfn, 2024.

[97] Aivin V Solatorio and Olivier Dupriez. Realtabformer: Generating realistic relational and tabular data using transformers, 2023.

[98] 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.

[99] Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramer, and Chiyuan Zhang. Quantifying memorization across neural language models, 2023.

[100] 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, editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 9782–9796, Miami, Florida, USA, November 2024. Association for Computational Linguistics.

[101] 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*, volume TBD, pages XXXX–YYYY, Vancouver, Canada, 2025. PMLR.

[102] Joshua Ward, Xiaofeng Lin, , Chi-Hua Wang, and Guang Cheng. Synth-mia: A testbed for auditing privacy leakage in tabular data synthesis, 2025.

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

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

[105] Li Yujian and Liu Bo. A normalized levenshtein distance metric. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(6):1091–1095, June 2007.

[106] 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, editors, *Proceedings of the 16th International Natural Language Generation Conference*, pages 28–53, Prague, Czechia, September 2023. Association for Computational Linguistics.

[107] Igor Shilov, Matthieu Meeus, and Yves-Alexandre de Montjoye. The mosaic memory of large language models, 2025.

[108] Jessup Byun, Xiaofeng Lin, Joshua Ward, and Guang Cheng. Risk in context: Benchmarking privacy leakage of foundation models in synthetic tabular data generation, 2025.

[109] Sebastian Felix Fischer, Matthias Feurer, and Bernd Bischl. OpenML-CTR23 – a curated tabular regression benchmarking suite. In *AutoML Conference 2023 (Workshop)*, Baltimore, MD, USA, 2023. PMLR.

[110] 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.

[111] 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.

[112] 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, 2025.

[113] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. Technical report, OpenAI, 2019.

[114] 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.

[115] 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.

[116] 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.

[117] Rakesh Agrawal and Jerry Kiernan. Watermarking relational databases. In *VLDB'02: Proceedings of the 28th International Conference on Very Large Databases*, pages 155–166, Hong Kong, China, 2002. Morgan Kaufmann.

[118] 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.

[119] Justus Mattern, Fatemehsadat Mireshghallah, Zhijing Jin, Bernhard Schoelkopf, Mrinmaya Sachan, and Taylor Berg-Kirkpatrick. Membership inference attacks against language models via neighbourhood comparison. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Findings of the Association for Computational Linguistics: ACL 2023*, pages 11330–11343, Toronto, Canada, July 2023. Association for Computational Linguistics.

[120] Filippo Galli, Luca Melis, and Tommaso Cucinotta. Noisy neighbors: Efficient membership inference attacks against LLMs. In Ivan Habernal, Sepideh Ghanavati, Abhilasha Ravichander, Vijayanta Jain, Patricia Thaine, Timour Igamberdiev, Niloofar Mireshghallah, and Oluwaseyi Feyisetan, editors, *Proceedings of the Fifth Workshop on Privacy in Natural Language Processing*, pages 1–6, Bangkok, Thailand, August 2024. Association for Computational Linguistics.

[121] Wenjie Fu, Huandong Wang, Chen Gao, Guanghua Liu, Yong Li, and Tao Jiang. Membership inference attacks against fine-tuned large language models via self-prompt calibration. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, NIPS '24, Red Hook, NY, USA, 2024. Curran Associates Inc.