Risk In Context: Benchmarking Privacy Leakage of Foundation Models in Synthetic Tabular Data Generation

Jessup Byun, Xiaofeng Lin, Joshua Ward, Guang Cheng
{jessupbyun,bernardo1998,joshuaward,guangcheng}@g.ucla.edu
University of California, Los Angeles
Los, Angeles, CA, USA

Abstract

Synthetic tabular data is essential for machine learning workflows, particularly as they expand small or imbalanced datasets and facilitate privacy-preserving data sharing. Yet, state-of-the-art generative models (GANs, VAEs, and diffusion models) achieve their promised fidelity only with large datasets containing thousands of examples. In low-data scenarios—often the primary motivation for using synthetic data—these models frequently overfit, leak sensitive records, and require constant retraining as new data arrive. To address these limitations, recent studies have leveraged large pre-trained transformers capable of generating new data rows via *in-context learning (ICL)*. These foundation models require only a few seed examples to produce additional records without any parameter updates, thus overcoming both data scarcity and retraining bottlenecks. However, in-context learning (ICL) repeats the seed rows verbatim at each generation step, creating a novel privacy threat that has so far been quantified only for text. How serious this leakage is for tabular synthesis—where a single row can uniquely identify an individual—remains unknown.

We close this gap with the first end-to-end benchmark of three foundation-model generators—GPT-4o-mini, LLaMA 3.3 70B, and TabPFN v2—against four state-of-the-art baselines on 35 real-world tables spanning health, finance, and public policy. Evaluated on statistical fidelity, downstream utility, and worst-case membership inference leakage, the study reveals that foundation models consistently occupy the upper end of the privacy-risk spectrum. We find that LLaMA 3.3 70B poses the highest privacy risk, yielding up to 54 percentage points (pp) higher true-positive rate (at 1% FPR) than the safest deep-learning baseline. GPT-4o-mini and TabPFN also rank among the most vulnerable models, revealing the elevated leakage potential of foundation models. Quality–leakage tradeoff plots illustrate a privacy–utility frontier, with CTGAN and GPT-4o-mini offering favorable balances. A factorial study demonstrates that three zero-cost prompt-level mitigations—small batch size, low (but non-zero) temperature, and inclusion of summary statistics—can reduce worst-case AUC by 14 pp and rare-class leakage by up to 39 pp, while retaining over 90% of baseline fidelity. Our benchmark and mitigation recipe provide an actionable blueprint for safer, low-data tabular synthesis using foundation models.

1 Introduction

Tabular data underpin a vast share of real-world machine-learning pipelines, from clinical registries to credit-risk engines. Consequently, *synthetic tabular data*—records drawn from a model that approximates the joint distribution of the original table—has become a key instrument for modern ML workflows [1]. Its value is two-fold. First, synthetic rows can augment *small or imbalanced datasets*, bolstering the statistical power of models trained on rare-disease cohorts or under-represented demographic groups. Second, because the generated rows do not correspond to real individuals, they enable data sharing while preserving privacy, a requirement that is particularly stringent in healthcare [2, 3] and finance [4]. By offering both stronger sample support and reduced disclosure risk [5, 6], synthetic tables now play a fundamental role in scaling machine learning to domains where real data are scarce or legally restricted.

However, deep table generators themselves are data-hungry learning algorithms. Empirical studies show that state-of-the-art GAN, VAE, and diffusion-based models for tabular data [7, 8, 9, 10, 11] attain their advertised fidelity only when thousands of training rows are available. In the very regimes where synthetic data are most demanded—rare-disease cohorts or minority-group slices with <500 real examples—these generators (i) overfit, reproducing near-duplicates that endanger privacy [12], and (ii) deliver little utility because the synthetic set lacks diversity [13]. Moreover, as production streams may append millions of new rows every hour, retraining such *data-specific* models becomes prohibitively slow [10], undermining their scalability in real deployments.

To overcome these limitations, researchers have turned to *large pre-trained transformers*—hereafter large language models (LLMs) and *foundation models for tabular data*. Such models, pre-trained on massive corpora, can generate new rows *in context*: given only a handful of seed examples, they sample additional records without any weight updates. This in-context learning (ICL) regularizes generation in low-data regimes, yielding realistic and diverse rows while eliminating the retraining bottleneck. Two representative families are (i) *general-purpose language models* prompted with column headers and a few exemplary rows [14, 15], and (ii) *tabular-specific transformers* pre-trained across thousands of small tables, such as TabPFN [16, 17].

Despite its promise, *in-context learning (ICL)* introduces significant privacy concerns. Specifically, the small number of real data rows embedded in the prompt are repeatedly exposed verbatim to the model during each generation step. Since the model undergoes no parameter updates, it is free to reproduce these seed rows token-by-token—a vulnerability recently documented in the context of large language models (LLMs). Wen *et al.* demonstrated that membership-inference attacks can achieve over 95% accuracy against several open-source LLMs, relying solely on observing generated outputs [18]. Additionally, Nakka *et al.* showed that multi-query attacks could increase the extraction rate of personally identifiable information by up to five times [19]. Even proposed methods that incorporate differential privacy directly into the prompts have

confirmed that naive ICL approaches remain susceptible to substantial row-level privacy leakage [20].

In the case of tabular data, the privacy risks are further exacerbated. Each record often represents a unique combination of quasi-identifiers—attributes such as rare disease diagnoses, ZIP codes, age, or income—which makes any verbatim reproduction immediately identifiable. However, a comprehensive and systematic evaluation quantifying the precise privacy risks associated with ICL specifically for synthetic tabular data generation has yet to be conducted.

Our contributions are as follows:

- **Holistic benchmark of privacy in low-data table synthesis.** We compare three foundation-model ICL generators (GPT-4o-mini, LLaMA 3.3 70B, and TabPFN v2) with four state-of-the-art data-specific baselines (CTGAN, TVAE, TabDiff, and SMOTE) on *35* real-world tables spanning health, finance, and public-policy domains. Evaluated on statistical fidelity, downstream utility, and worst-case membership-inference leakage, the study reveals that foundation models occupy the upper end of the privacy-risk spectrum: Llama 3.3 70B yields up to **54%** higher true-positive rate (at 1% FPR) than the safest deep-learning baseline, while GPT-4o-mini and TabPFN sit near the high end of the deep-generator cluster. These cross-model, cross-dataset leakage patterns have not been quantified before.
- **Prompt-level privacy knobs and trade-off frontier.** A factorial study of three zero-cost prompt parameters—(i) generation batch size, (ii) sampling temperature, and (iii) inclusion of summary statistics—maps how simple edits shift foundation models along the privacy–utility frontier. Tuning these levers cuts worst-case AUC by up to 14 pp and slashes rare-class leakage by 24–39 pp, while preserving >90% of baseline fidelity. The recipe of *small batch, low (but non-zero) temperature, and explicit summary statistics* is an immediately deployable defense that requires no retraining.

2 Related Work

2.1 Synthetic Tabular Data Generation

Early work on STDG was rooted in classical statistical techniques such as CART synthesis, parametric bootstrap, and SMOTE-based resampling [21, 22, 23]. While effective for small, low-dimensional tables, these approaches struggle to capture complex multivariate dependencies.

Deep generative models. Recent deep-learning methods have markedly expanded the scope of STDG [24, 25]. CTGAN and TVAE [7] marry conditional generation with Generative Adversarial Networks (GANs) and Variational Autoencoders (VAEs) to cope with imbalanced, heavy-tailed columns. CtabGAN+ [26, 9] extends this idea to mixed-type variables with long-tailed distributions, whereas *TabDDPM* [10] achieves state-of-the-art fidelity by running separate diffusion processes for numerical versus categorical features. *Autodiff* [27] and *Tabsyn* [11] follow a similar diffusion paradigm but rely on table-specific autoencoders, limiting their cross-domain transferability.

Privacy-aware generators. A parallel line of work incorporates Differential Privacy (DP) into the training objective [28, 29, 6]. Although DP provides formal guarantees, existing DP generators often sacrifice either statistical fidelity or downstream utility, failing to improve model performance when the synthetic data are used for augmentation [30].

Foundation-model approaches. Prompt-fine-tuning methods such as *GReaT* [31] and *Tabula* [32] first linearize each row into natural-language text and then *continue training* the language model. Although effective, they require gradient updates and large GPU budgets.

A parallel line of work keeps the backbone *frozen* and relies purely on *in-context learning*. Curated LLM [15] generates synthetic tables in low-data regimes combining prompting LLM API with example rows and column statistics, and curating feature relationships in synthesized tables via learning dynamics of oracle models. Kim *et al.* introduces EPIC [33], a grouped-CSV prompting scheme that lets an off-the-shelf LLM synthesize balanced data for minority classes without any fine-tuning. For tabular-specific transformers, TabPFN v2 [17] performs tabular synthesis by sequential in-context prediction of the next feature given generated ones. Ma *et al.* turns the pre-trained TabPFN into an energy-based generator dubbed TabPFGen [34]; sampling is performed with Langevin dynamics, again requiring no additional training. These ICL-only techniques demonstrate that foundation models can act as drop-in tabular generators, yet their privacy characteristics remain largely unexplored.

Overall, while modern deep generative models significantly outperform classical statistical methods, they remain impractical due to the constant retraining required for each new dataset. Meanwhile, foundation-model approaches present new challenges, particularly numerical precision issues and increased privacy risks—these are precisely the gaps our work addresses.

2.2 Privacy Auditing of Tabular Synthetic Data With Membership-Inference Attack

Membership Inference Attacks (MIAs) aim to classify whether a specific observation was a member of the original dataset used to train a model. Let X be a random variable on domain $\mathcal{X}$ with distribution $p_X(X)$, and T be a dataset of independent samples from $p_X(X)$. A generative model G, trained on T, then generates a synthetic dataset S. An adversary $\mathcal{A} : X \to \{0, 1\}$ aims to determine if a test sample x^* is an element of T: $T \sim p_X(X)$. Formally, this classification or Membership Inference Attack can be expressed as:

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

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

To construct their attack, the adversary relies on some prior given information called a threat model. These include black box attacks [35, 36, 37] in which only S is available, shadow box (also called calibrated) attacks in which both S and then a reference dataset R from the same population distribution of the training set are given [12, 38], and white box attacks [39] in which both S, R and full access to the model are known. Other lines of work have

explored threat models where the adversary assumes a shadow-box threat model but additionally knows the implementation, but not the training weights, of the tabular generator [40, 41, 42].

MIAs leverage information from a specified threat model along with some observation regarding model failure behavior to exploit potential vulnerabilities in constructing Equation 1. For example, a variety of attacks from [37] and [41] target memorization by computing the distance between x^* and the closest observation from S. Other MIAs focus on overfitting, where the model produces synthetic samples that are too similar in distribution to the training dataset relative to the overall population distribution. Methods such as DOMIAS [12] and DPI [38] attack overfitting by comparing the density of synthetic observations in a local region to that of a reference dataset.

While methodologically diverse, MIAs targeting synthetic data aim to uncover the same fundamental issue: the potential for generative models to inadvertently reveal information about their training data. If a model produces synthetic records that allow an adversary to infer training membership, it constitutes a direct breach of privacy. This leakage signals a failure in the model, as it indicates an imbalance between generating realistic data and preserving confidentiality. A well-calibrated generative model should neither reproduce training samples nor generate synthetic data that is overly concentrated around specific regions of the training distribution.

3 Methodology

Our benchmark evaluates a spectrum of tabular–data generators under a unified low–data protocol. Below we detail (i) the models under comparison, (ii) the dataset and sampling procedure, and (iii) the evaluation metrics used to quantify fidelity, downstream utility, and privacy leakage.

3.1 Synthetic–Data Generators

(A) Foundation models *via* in-context learning (ICL).

- **TabPFN v2** [17]: a transformer pre-trained across millions of synthetic tables for few-shot tabular prediction. We adapt it to generation by autoregressively sampling each feature conditioned on previously sampled features.
- **GPT-4o mini** [43]: a general-purpose LLM queried using a structured prompt that specifies the dataset name, schema (column names and ordering), statistical summaries (numerical and categorical), and a sample of the real data. The prompt includes a complete CSV-formatted snippet of the training data and instructs the model to generate new samples with matching structure and diversity. To encourage faithful yet varied outputs, we also describe the generation task as mimicking causal and statistical properties while explicitly avoiding extra columns or formatting artifacts. The full prompt is detailed in Appendix B.
- **LLaMA 3.3 70B** [44]: an open-source LLM prompted analogously to GPT-4o mini, enabling head-to-head comparison between closed-and open-source LLMs.

(B) Data-specific generators (trained per dataset).

- **TabDiff** [45]: a joint continuous-time diffusion model for mixed-type tabular data that defines feature-wise learnable diffusion processes to capture the heterogeneity of numerical and categorical columns.
- **CTGAN** and **TVAE** [7]: CTGAN employs a conditional GAN with mode-specific normalization layers using a Gaussian mixture model to encode continuous features and conditional vectors for categorical features, plus "training-by-sampling" to improve mode coverage; TVAE uses a variational autoencoder to embed mixed-type features into a continuous latent space and decodes via mode-specific normalization, optimizing the ELBO with structural regularization to balance fidelity and diversity.
- **SMOTE** [22]: a k-nearest-neighbor oversampling technique that generates synthetic minority-class samples by interpolating between each sample and its randomly selected nearest neighbors in feature space.

3.2 Datasets and Low-Data Protocol

Public benchmark. We use the OpenML CTR23 suite [46], comprising of 35 classification tables with heterogeneous schema. The suite comprises 35 real-world tables spanning 500–100,000 rows and no more than 5000 engineered features after one-hot encoding; numerical attributes dominate, but every dataset also includes several categorical columns, giving the heterogeneous structure that we target.

Splitting and subsampling. For each dataset, we create an $80:20$ train–test split on the real data, stratified based on target classes. To emulate low-data regimes, we draw $\{32, 64, 128\}$ training rows without replacement and repeat the draw with random seeds $0, 1, 2$. ICL models receive the sampled rows as exemplars; trainable generators fit their parameters on the same subsample.

3.3 Evaluation Metrics

We benchmark every generator along three complementary axes: (*i*) statistical fidelity of the synthetic table, (*ii*) downstream utility in real predictive tasks, and (*iii*) privacy leakage against membership-inference attacks (MIAs).

3.3.1 Statistical Fidelity. For statistical fidelity, we measure both the marginal column distribution as well as joint distribution. For numerical columns, we compute the Kolmogorov–Smirnov (KS) distance between real and synthetic marginals; for categorical columns, we use the χ^2 divergence on contingency tables. We subtract the column distance from 1 so that larger values indicate smaller distance and thus better marginal distribution modeling. For joint distribution, we use similarity of column correlation in real and synthetic data. We calculate Pearson's correlation for numerical-numerical column pairs; for categorical-categorical pairs, we compute a normalized contingency table for the real and synthetic data. This table describes the proportion of rows that have each combination of categories in A and B. Then, it computes the difference between the contingency tables using the Total Variation Distance. Finally, we subtract the distance from 1 to ensure that a high score means high similarity. For numerical-categorical pairs, we convert the numerical column into bins by quantile and compute the contingency similarity. We followed the implementation of SDMetrics [47] of fidelity metrics.

3.3.2 Downstream Utility. To evaluate the machine learning utility of synthetic data, we fit the following classifiers: logistic regression, Naïve Bayes, decision tree, random forest, XGBoost [48], and CatBoost [49], and then evaluate them on the holdout test set with a macro-average ROC AUC score. For

3.3.3 Privacy Leakage.

Threat model. We consider black-box and model-unknown shadow-box adversaries who see the released synthetic set S— optionally supplemented by a reference set R— but never the generator code or parameters. This mirrors realistic data-sharing scenarios and enables model-agnostic, computationally feasible audits [12, 50].

Auditing procedure. Following the empirical-worst-case principle of Empirical Differential Privacy (EDP) [51], we measure leakage as the *maximum* attack AUC across $\mathcal{A} = 13$ state-of-the-art MIAs that span distance, density, and classifier signals:

$$\text{Leakage} = \max_{A \in \mathcal{A}} \text{AUC}(A).$$

The Area Under Receiver Operating Characteristic curve (ROC AUC) and True Positive Rate (TPR) at False Positive Rate (FPR) are used as performance metrics of attacks, where a higher performance indicates more success in identifying membership of training data points, and thus greater privacy leakage. All attacks are re-implemented in a common Python framework; no model re-training is required. Table 1 lists the methods. We used the Synth-MIA [52] library as our attack framework implementation.

Table 1: Membership-inference attacks used in this study.

Attack	Threat model	Signal type
DOMIAS [12]	Shadow-box	Density ratio
DPI [38]	Shadow-box	Local density
Classifier [41]	Shadow-box	Density ratio
Density Estimator [41]	Black-box	Density estimation
DCR [37]	Black-box	Distance-based
DCR-Diff [37]	Shadow-box	Distance difference
Logan [35]	Shadow-box	Density ratio
MC Estimation [36]	Black-box	Density estimation

4 Experiments

4.1 Privacy Leakage

Table 2 reports the *worst-case* membership-inference performance, averaged over all datasets, subset sizes, and random seeds—for each generator class. Two consistent patterns emerge:

Foundation models sit at the higher end of the privacy-risk spectrum. The three foundation-model variants occupy the upper end of the leakage range: their mean worst-case AUCs fall in the range 0.587–0.667, compared with 0.580–0.627 for CTGAN, TVAE, and TabDiff. At an operational false-positive rate of 1%, the corresponding $\text{TPR}_{0.01}$ climbs from 0.042–0.061 (deep generators) to 0.054–0.181 (foundation models), indicating that an adversary can recover up to three times as many true members with high confidence when attacking a foundation-model synthesizer. Nonetheless, all deep-learning generators remain substantially safer than the interpolation-based baseline SMOTE, whose AUC of 0.831 and $\text{TPR}_{0.01} = 0.438$ confirm that naïve interpolation-based upsampling of minority rows is highly vulnerable.

Large language models vary widely in leakage. Within the foundation-model family, leakage is *not uniform*. LLaMA 3.3 70B shows the highest risk (AUC 0.667, $\text{TPR}_{0.1} = 0.345$), exceeding even the safest data-specific generator by roughly +0.09 pp. By contrast, TabPFN v2 (0.620) and GPT-4o mini (0.587) hover at the upper end of the deep-generator cluster, suggesting that model architecture, prompt format, or pre-training corpus strongly influences membership exposure. This disparity underscores the need for per-model audits rather than blanket assumptions about "foundation-model risk."

4.2 Impact of Dataset Size

Figure 2 illustrates how the mean worst-case AUC varies when the number of real rows supplied to each generator is increased from $n = 32$ to $n = 128$. Three key observations stand out:

(1) **Smaller samples leak more.** Every *learned* generator—GAN, VAE, diffusion, and foundation model—shows a monotone decrease in leakage as the training subset grows. For instance, TVAE drops from 0.668 ($n = 32$) to 0.587 ($n = 128$), while GPT-4o mini falls from 0.617 to 0.563. The trend supports the intuition that with fewer examples the model must rely on memorisation, thereby exposing members more readily.

(2) **Foundation-model dispersion persists across sizes.** Although leakage lessens with data, the ranking among foundation models remains: LLaMA 3.3 70B is consistently the riskiest (AUC 0.713 → 0.625), whereas TabPFN v2 and GPT-4o mini track the upper bound of the deep-generator cluster.

(3) **SMOTE is size-insensitive yet unsafe.** The interpolation baseline stays near 0.83 AUC regardless of n, confirming that simply copying or perturbing minority rows offers no privacy benefit even when more seeds are available.

Implication. Low-data scenarios—precisely the cases where synthetic augmentation is most needed—also pose the greatest privacy exposure, particularly for large LLMs. Mitigation efforts should therefore prioritize these extreme regimes.

4.3 Trade-off between Privacy and Data Quality

Privacy–Utility Trade-off. Figure 2 juxtaposes the mean worst-case privacy AUC (x-axis) with four quality indicators (y-axis): (column-pair) *correlation similarity*, *column shape similarity*, downstream *classifier AUC*, and *beta recall* [53] (diversity). Across all panels we observe a clear, approximately monotone frontier: models that score higher on any quality metric also suffer larger membership-inference leakage. Put differently, *better data quality lines up with weaker privacy.*

Deviation from the naive $y = x$ frontier. In an ideal scenario where each unit of quality gain incurs an equal privacy cost, generators would follow a diagonal "naive frontier" resembling a $y = x$ line. In practice, however, models deviate from this pattern: CTGAN and GPT-4o-mini deliver moderate quality with relatively low leakage—offering better trade-offs and falling *below* this idealized trend.

Table 2: Privacy leakage metrics: mean (standard deviation) of the worst-case attacker performance across all dataset, subset sizes and seeds.

Model	AUC	TPR@FPR=0	TPR@FPR=0.001	TPR@FPR=0.01	TPR@FPR=0.1
SMOTE	0.831 (0.064)	0.273 (0.211)	0.310 (0.202)	0.438 (0.181)	0.648 (0.136)
CTGAN	0.580 (0.047)	0.008 (0.016)	0.012 (0.018)	0.042 (0.028)	0.185 (0.058)
TVAE	0.627 (0.065)	0.017 (0.033)	0.023 (0.034)	0.061 (0.048)	0.244 (0.087)
TabDiff	0.600 (0.062)	0.012 (0.024)	0.016 (0.024)	0.052 (0.038)	0.213 (0.083)
GPT-4o-mini	0.587 (0.052)	0.016 (0.027)	0.023 (0.030)	0.054 (0.038)	0.205 (0.074)
TabPFN v2	0.620 (0.059)	0.015 (0.025)	0.023 (0.028)	0.060 (0.038)	0.243 (0.078)
LLaMA 3.3 70B	**0.667** (0.128)	**0.100** (0.224)	**0.117** (0.236)	**0.181** (0.264)	**0.345** (0.240)

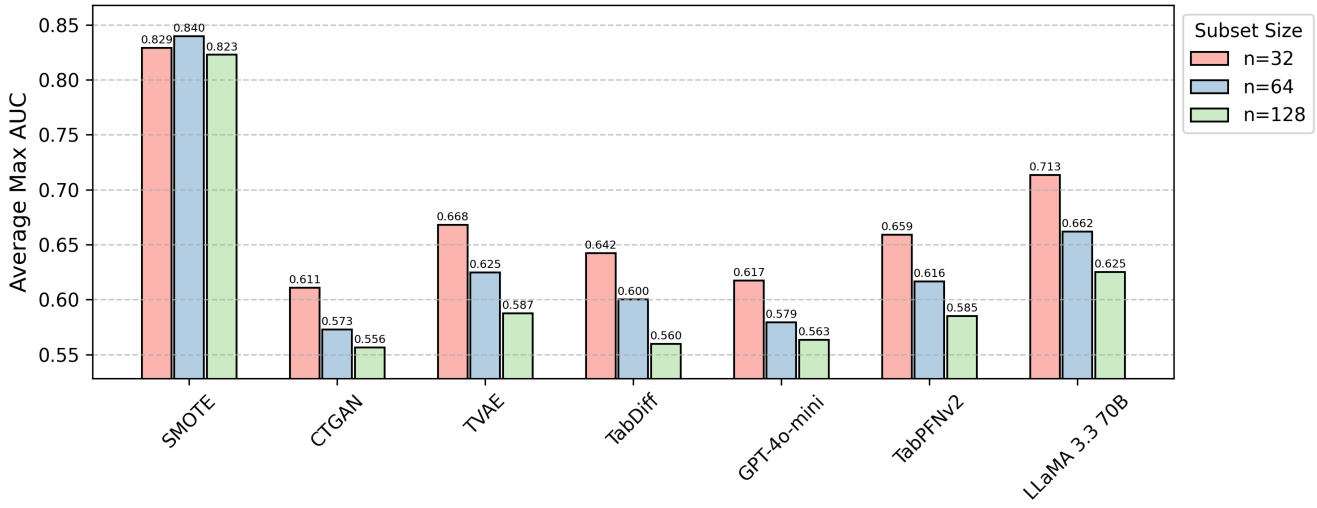


Figure 1: Privacy leakage (mean worst-case AUC) for each synthetic data generator across three subset sizes. For each model we compute the maximum AUC over all attackers for each of the 315 splits (35 datasets × 3 seeds × 3 sizes), then show the mean of those 315 worst-case values. Bars are color-coded by subset size ($n = 32, 64, 128$).

In contrast, LLaMA 3.3 70B and especially SMOTE incur significantly higher privacy risks for their utility gains, landing *above* the trend. These deviations reveal that better privacy–utility trade-offs are achievable, and that there is clear headroom for improvement.

Take-away: Foundation-model in-context synthesis can match or exceed traditional generators in utility, but this comes at a measurable privacy cost—particularly for open-source LLMs such as LLaMA 3.3 70B—reinforcing our call for lightweight leakage-mitigation practices.

4.4 Factorial Study of Privacy Drivers

Findings and implications. Table 3 probes four single–parameter tweaks to the LLaMA-3.3 70B prompting pipeline on 4 dataset with the highest average max AUC(strongest privacy leakage).

- **Smaller generation batch (10 rows).** Reducing the batch from its 100-row default lowers the worst-case attack AUC by 14 pp and slashes the *rare-class* AUC from 0.979 to 0.740. The price is an 11 pp drop in marginal-shape similarity and an almost complete collapse of correlation similarity, showing that privacy gains come mainly from a loss of diversity.
- **Removing summary statistics from the prompt.** Omitting global means/variances *increases* leakage to 0.878 (+0.064) while also hurting fidelity, indicating that summary statistics act as a soft regularizer anchoring the model to population-level trends.
- **Sampling temperature.** Lowering the temperature to 0.1 trims AUC only marginally (0.814 → 0.811) but still reduces rare-class exposure (0.979 → 0.842). Conversely, raising it to 0.5 nudges leakage up to 0.818 yet yields the best correlation similarity (0.258), highlighting temperature as a fine-grain knob on the privacy–utility frontier.

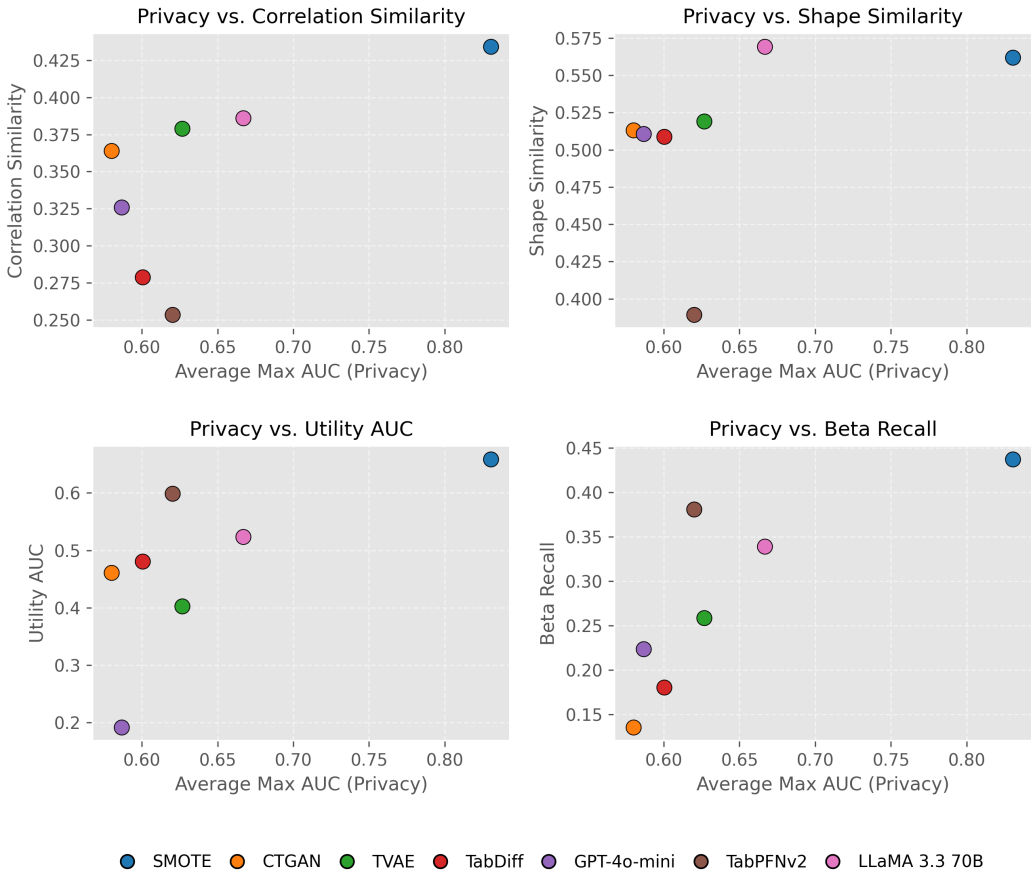
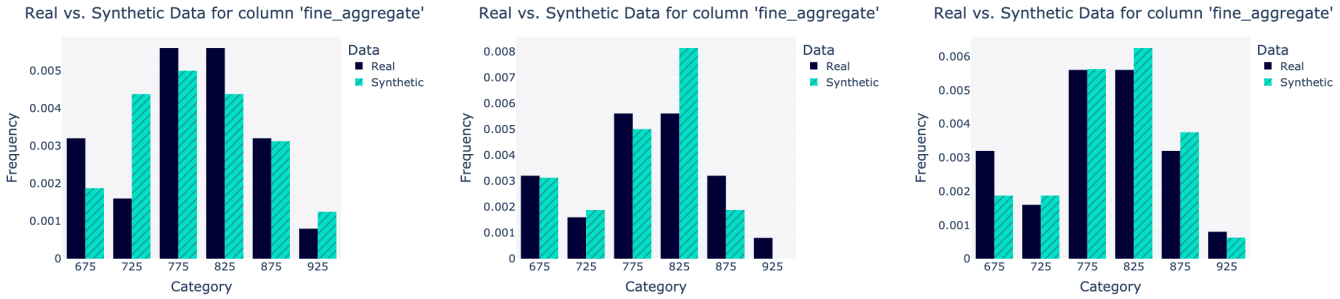


Figure 2: Privacy–utility trade-off across synthetic generators. Each point represents one generator's *mean* worst-case membership-inference AUC (x-axis: Average Max AUC, Privacy) plotted against its *mean* fidelity/diversity/utility score (y-axis: metrics), for correlation similarity, shape similarity, utility AUC, and beta recall in a 2×2 grid. Points are colored by model; legend at bottom. Error bars omitted for clarity.

Table 3: Factorial Study of Privacy Drivers Results. Each block shows the mean over four datasets with the highest average max AUC (concrete-compressive-strength, naval-propulsion-plant, solar-flare, white-wine) of the maximum attack AUC, average column-shape similarity, column correlation similarity, proportion of synthetic samples closer to real, and rare-class ROC AUC.

Ablation	Max Attack AUC	Avg. Shape	Corr Similarity	Prop. Closer	Rare-Class AUC
Default	0.814	0.380	0.232	0.784	0.979
Batch size (k=10)	0.702	0.340	0.036	0.727	0.740
Summary stats not in prompt	0.878	0.365	0.200	0.783	0.904
Temperature = 0.1	0.811	0.364	0.222	0.758	0.842
Temperature = 0.5	0.818	0.392	0.258	0.779	0.860

Why does attack efficacy *fall* despite fewer exemplars and lower randomness? At first glance, one might expect that conditioning on fewer rows (smaller batch) or sampling with a near-deterministic decoder (low temperature) would *increase* lead to increased leakage: the model has less contextual diversity and may simply copy the in-context examples. Our results show the opposite, and the key lies in how attacks exploit *rare values*. Membership-inference methods rely most heavily on rows whose quasi-identifiers



(a) Default (batch = 32, temp = 1.0) **(b) Reduced batch (batch = 10, temp = 1.0)** **(c) Low temperature (batch = 32, temp = 0.1)**

Figure 3: Binned value–frequency distributions for the numeric column `fine_aggregate` from one of the 35 benchmark datasets in our CTR-23 suite, under three prompt settings for a single split ($n = 32$, seed 0), generated by LLaMA 3.3 70B. Each panel shows the frequency of rows falling into equal-width bins of the original variable. Comparing (a) default prompt, (b) small batch size, and (c) low temperature, highlights how these factors concentrate mass in central bins and suppresses long-tail diversity.

are infrequent in the population; such outliers maximize the attacker's posterior shift when observed in the synthetic set. By contrast, high-frequency tuples offer little discriminative signal. Both knobs we vary push the generator toward the mode of the data distribution:

- **Smaller batch size** reduces the chance that any rare combination even appears in the prompt, biasing the model toward majority patterns seen across tasks.
- **Lower temperature** shrinks the softmax entropy at each decoding step, further concentrating mass on high-probability categories or central numeric quantiles.

Visualizing diversity collapse. Figure 3 presents per-category frequency histograms under three conditions: (a) the default prompt, (b) reduced batch size, and (c) low sampling temperature. Both ablations exhibit a pronounced loss of support in the distributional tails—rare bins become markedly underrepresented—while the default configuration preserves coverage across the full range of real values. This contraction of long-tail coverage closely parallels the observed decline in rare-class AUC, reinforcing the conclusion that privacy improvements arise primarily from the suppression of low-frequency events rather than from wholesale elimination of copying common patterns.

To verify this mechanism, we partition each training split into *rare* and *common* subsets, labelling a value as rare if its category frequency is $\leq 5\%$ (categorical) or if a numeric binned into 20 equal-width intervals also falls into a $\leq 5\%$ bin. We then compute the worst-case AUC *restricted to rare rows*. As shown in the rightmost column of Table 3, the rare-class AUC plummets from 0.979 (default) to 0.740 (batch = 10) and 0.842 (temperature = 0.1)—drops of 24–39 percentage points that dwarf the 3–11 percentage changes in overall AUC. The privacy gain therefore stems almost entirely from suppressing rare-value emission rather than from any reduction in verbatim copying of common rows. This finding reframes the trade-off: *privacy is improved because diversity, particularly on the long tail, is curtailed*, underscoring a fundamental tension between protecting outliers and preserving their statistical signal.

Practical takeaway. Prompt-level controls—batch size, inclusion of summary statistics, and temperature—let practitioners traverse the privacy–utility trade-off *without retraining*. A conservative default for sensitive releases is: *small batch, low (but non-zero) temperature, and explicit summary statistics*, which cuts worst-case AUC by up to 0.11 while keeping marginal-shape similarity within 10% of the baseline.

5 Conclusion

We have presented the first systematic, end-to-end benchmark of foundation-model in-context tabular generators against data-specific GAN, VAE, and diffusion baselines under low-data regimes. Evaluating 35 public datasets with split sizes of 32, 64, and 128 real rows, we measured statistical fidelity, downstream utility, and worst-case membership-inference leakage across 13 state-of-the-art attacks. Our results reveal that while foundation models could achieve competitive fidelity and predictive performance, they also occupy the upper end of the privacy-risk spectrum. Simple prompt-level adjustments (batch size, summary-stat inclusion, and temperature) can shift a generator along the privacy–utility frontier by cutting leakage up at modest fidelity cost, without any retraining.

Despite these insights, our study has several limitations. We focus solely on membership-inference as the privacy metric; other risks (attribute inference, reconstruction attacks, or linkage to auxiliary data) remain unquantified. Second, our prompt designs and hyper-parameters cover a limited slice of the vast LLM configuration space, and future work should explore automated prompt tuning or private prompting methods. Addressing these gaps—by broadening attack models, integrating formal differential-privacy guarantees, and extending benchmarks to richer data domains—will be crucial to deploying foundation models safely in privacy-sensitive applications. As organizations increasingly turn to generative tools for data augmentation and sharing, our benchmark offers a foundational template for evaluating trade-offs between data fidelity, utility, diversity, and privacy risk in realistic, low-data contexts.

References

[1] Joao Fonseca and Fernando Bacao. 2023. Tabular and latent space synthetic data generation: a literature review. *Journal of Big Data*, 10, 1, 115.

[2] Vibeke Binz Vallevik et al. 2024. Can i trust my fake data–a comprehensive quality assessment framework for synthetic tabular data in healthcare. *International Journal of Medical Informatics*, 105413.

[3] Mikel Hernandez, Gorka Epelde, Ane Alberdi, Rodrigo Cilla, and Debbie Rankin. 2022. Synthetic data generation for tabular health records: a systematic review. *Neurocomputing*, 493, 28–45.

[4] Jinhong Wu, Konstantinos Plataniotis, Lucy Liu, Ehsan Amjadian, and Yuri Lawryshyn. 2023. Interpretation for variational autoencoder used to generate financial synthetic tabular data. *Algorithms*, 16, 2, 121.

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

[6] Ryan McKenna, Brett Mullins, Daniel Sheldon, and Gerome Miklau. 2022. Aim: an adaptive and iterative mechanism for differentially private synthetic data. *Proceedings of the VLDB Endowment*, 15, 11, 2599–2612.

[7] Lei Xu, Maria Skoularidou, Alfredo Cuesta-Infante, and Kalyan Veeramachaneni. 2019. Modeling tabular data using conditional gan. *Advances in neural information processing systems*, 32.

[8] Jayoung Kim, Jinsung Jeon, Jaehoon Lee, Jihyeon Hyeong, and Noseong Park. 2021. Oct-gan: neural ode-based conditional tabular gans. In *Proceedings of the Web Conference 2021*, 1506–1515.

[9] Zilong Zhao, Aditya Kunar, Robert Birke, Hiek Van der Scheer, and Lydia Y Chen. 2024. Ctab-gan+: enhancing tabular data synthesis. *Frontiers in big Data*, 6, 1296508.

[10] Akim Kotelnikov, Dmitry Baranchuk, Ivan Rubachev, and Artem Babenko. 2023. Tabddpm: modelling tabular data with diffusion models. In *International Conference on Machine Learning*. PMLR, 17564–17579.

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

[12] Boris van Breugel, Hao Sun, Zhaozhi Qian, and Mihaela van der Schaar. 2023. Membership inference attacks against synthetic data through overfitting detection. (2023). arXiv: 2302.12580 [cs.LG].

[13] Reilly Cannon, Nicolette M. Laird, Caesar Vazquez, Andy Lin, Amy Wagler, and Tony Chiang. 2025. Assessing generative models for structured data. *arXiv preprint arXiv:2503.20903*. https://arxiv.org/abs/2503.20903.

[14] Zhuoyan Li, Hangxiao Zhu, Zhuoran Lu, and Ming Yin. 2023. Synthetic data generation with large language models for text classification: potential and limitations. *arXiv preprint arXiv:2310.07849*.

[15] Nabeel Seedat, Nicolas Huynh, Boris van Breugel, and Mihaela van der Schaar. 2024. Curated llm: synergy of llms and data curation for tabular augmentation in low-data regimes. In *Forty-first International Conference on Machine Learning*.

[16] Noah Hollmann, Samuel Müller, Katharina Eggensperger, and Frank Hutter. 2022. Tabpfn: a transformer that solves small tabular classification problems in a second. *arXiv preprint arXiv:2207.01848*.

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

[18] Rui Wen, Zheng Li, Michael Backes, and Yang Zhang. 2024. Membership inference attacks against in-context learning. In *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*. ACM, 1–15.

[19] Krishna Kanth Nakka, Ahmed Frikha, Ricardo Mendes, Xue Jiang, and Xuebing Zhou. 2025. PII-scope: a comprehensive study on training data PII extraction attacks in large language models. *arXiv preprint arXiv:2410.06704*.

[20] Xinyu Tang, Richard Shin, Huseyin A. Inan, Andre Manoel, Fatemehsadat Mireshghallah, Zinan Lin, Sivakanth Gopi, Janardhan Kulkarni, and Robert Sim. 2024. Privacy-preserving in-context learning with differentially private few-shot generation. In *International Conference on Learning Representations (ICLR)*.

[21] Jerome P Reiter. 2005. Using cart to generate partially synthetic public use microdata. *Journal of official statistics*, 21, 3, 441.

[22] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. 2002. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16, 321–357.

[23] Beata Nowok, Gillian M Raab, and Chris Dibben. 2016. Synthpop: bespoke creation of synthetic data in r. *Journal of statistical software*, 74, 1–26.

[24] Zhengping Che, Yu Cheng, Shuangfei Zhai, Zhaonan Sun, and Yan Liu. 2017. Boosting deep learning risk prediction with generative adversarial networks for electronic health records. In *2017 IEEE International Conference on Data Mining (ICDM)*. IEEE, 787–792.

[25] Alvaro Figueira and Bruno Vaz. 2022. Survey on synthetic data generation, evaluation methods and gans. *Mathematics*, 10, 15, 2733.

[26] Zilong Zhao, Aditya Kunar, Robert Birke, and Lydia Y Chen. 2021. Ctab-gan: effective table data synthesizing. In *Asian Conference on Machine Learning*. PMLR, 97–112.

[27] Namjoon Suh, Xiaofeng Lin, Din-Yin Hsieh, Merhdad Honarkhah, and Guang Cheng. 2023. Autodiff: combining auto-encoder and diffusion model for tabular data synthesizing. *CoRR*, abs/2310.15479. https://doi.org/10.48550/arXiv.2310.15479.

[28] James Jordon, Jinsung Yoon, and Mihaela Van Der Schaar. 2018. Pate-gan: generating synthetic data with differential privacy guarantees. In *International conference on learning representations*.

[29] Jun Zhang, Graham Cormode, Cecilia M Procopiuc, Divesh Srivastava, and Xiaokui Xiao. 2017. Privbayes: private data release via bayesian networks. *ACM Transactions on Database Systems (TODS)*, 42, 4, 1–41.

[30] Dionysis Manousakas and Sergül Aydöre. 2023. On the usefulness of synthetic tabular data generation. *arXiv preprint arXiv:2306.15636*.

[31] Vadim Borisov, Kathrin Sessler, Tobias Leemann, Martin Pawelczyk, and Gjergji Kasneci. 2023. Language models are realistic tabular data generators. In *The Eleventh International Conference on Learning Representations*. https://openreview.net/forum?id=cEygmQNOeI.

[32] Zilong Zhao, Robert Birke, and Lydia Chen. 2023. Tabula: harnessing language models for tabular data synthesis. *arXiv preprint arXiv:2310.12746*.

[33] Jinhee Kim, Taesung Kim, and Jaegul Choo. 2025. Epic: effective prompting for imbalanced-class data synthesis in tabular data classification via large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*. arXiv:2404.12404.

[34] Junwei Ma, Apoorv Dankar, George Stein, Guangwei Yu, and Anthony Caterini. 2024. Tabpfgen – tabular data generation with tabpfn. *arXiv preprint arXiv:2406.05216*.

[35] Jamie Hayes, Luca Melis, George Danezis, and Emiliano De Cristofaro. 2017. Logan: membership inference attacks against generative models. *Proceedings on Privacy Enhancing Technologies*, 2019, 133–152. https://api.semanticscholar.org/CorpusID:52211986.

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

[37] Dingfan Chen, Ning Yu, Yang Zhang, and Mario Fritz. 2020. 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, (Oct. 2020). doi:10.1145/3372297.3417238.

[38] Joshua Ward, Chi-Hua Wang, and Guang Cheng. 2024. Data plagiarism index: characterizing the privacy risk of data-copying in tabular generative models. *KDD- Generative AI Evaluation Workshop*. https://arxiv.org/abs/2406.13012 arXiv: 2406.13012 [cs.LG].

[39] Alexandre Sablayrolles, Matthijs Douze, Cordelia Schmid, Yann Ollivier, and Hervé Jégou. 2019. White-box vs black-box: bayes optimal strategies for membership inference. In *International Conference on Machine Learning*. PMLR, 5558–5567.

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

[41] Florimond Houssiau, James Jordon, Samuel N Cohen, Owen Daniel, Andrew Elliott, James Geddes, Callum Mole, Camila Rangel-Smith, and Lukasz Szpruch. 2022. Tapas: a toolbox for adversarial privacy auditing of synthetic data. *arXiv preprint arXiv:2211.06550*.

[42] 2024. *Achilles' heels: vulnerable record identification in synthetic data publishing*. *Lecture Notes in Computer Science*. Springer Nature Switzerland, 380–399. ISBN: 9783031514760. doi:10.1007/978-3-031-51476-0_19.

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

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

[45] Juntong Shi, Minkai Xu, Harper Hua, Hengrui Zhang, Stefano Ermon, and Jure Leskovec. [n. d.] Tabdiff: a unified diffusion model for multi-modal tabular data generation. In *NeurIPS 2024 Third Table Representation Learning Workshop*.

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

[47] DataCebo, Inc. 2023. *Synthetic Data Metrics*. Version 0.12.0. DataCebo, Inc. (Oct. 2023). https://docs.sdv.dev/sdmetrics/.

[48] Tianqi Chen and Carlos Guestrin. 2016. Xgboost: a scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 785–794.

[49] Liudmila Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin. 2018. Catboost: unbiased boosting with categorical features. *Advances in neural information processing systems*, 31.

[50] Steven Golob, Sikha Pentyala, Anuar Maratkhan, and Martine De Cock. 2024. Privacy vulnerabilities in marginals-based synthetic data. (2024). https://arxiv.org/abs/2410.05506 arXiv: 2410.05506 [cs.CR].

[51] Matthew Jagielski, Jonathan Ullman, and Alina Oprea. 2020. Auditing differentially private machine learning: how private is private sgd? In *Proceedings of the 34th International Conference on Neural Information Processing Systems* (NIPS '20) Article 1862. Curran Associates Inc., Vancouver, BC, Canada, 12 pages. ISBN: 9781713829546.

[52] Joshua Ward, Xiaofeng Lin, Chi-Hua Wang, and Guang Cheng. Synth-MIA: A Testbed for Auditing Privacy Leakage in Tabular Data Synthesis. Manuscript under review, Los Angeles, CA, USA, (2025).

[53] Ahmed Alaa, Boris Van Breugel, Evgeny S Saveliev, and Mihaela van der Schaar. 2022. How faithful is your synthetic data? sample-level metrics for evaluating and auditing generative models. In *International Conference on Machine Learning*. PMLR, 290–306.

A Generator Implementation Details

TabPFN Generation [17]: We follow the Prior Labs tutorial (https://priorlabs.ai/tutorials/unsupervised/). Each train split is loaded, shuffled, and batched (200 rows). Numeric features are cast to `float32`, and categoricals are label-encoded (unseen → −1). Zero-variance columns are dropped before fitting and reinserted after sampling. For each batch, we fit the unsupervised TabPFN model and sample synthetic rows with temperature $t = 1.0$ across three random permutations. Outputs are decoded, constants reattached, batches concatenated, and truncated to the original row count.

LLaMA Generation [44]: Using LLaMA 3.3 70B via the Groq API (https://console.groq.com/docs/models), each train split is divided into batches of up to 32 rows. We ensure all rows are included in the prompt. Summary statistics are computed per column, and the batch is serialized to CSV. We call `llama-3.3-70b-versatile` with temperature=1.0, requesting N rows as JSON. If parse errors or incorrect lengths occur, we retry up to 5 times. Valid outputs are concatenated, truncated, or re-prompted, and validated for type and dimension consistency.[1]

GPT-4o-mini [43]: Used the same prompt and inference pipeline as LLaMA 3.3 70B. We use the structured output API, with the format defined as a JSON schema where keys are column names and values are cell entries.

SMOTE: We extend the original SMOTE algorithm to interpolate all classes. A target class is randomly sampled based on empirical frequencies, followed by interpolation using the $k = 5$ nearest neighbors and $\alpha = 0.5$.

CTGAN: We use the official implementation at https://github.com/sdv-dev/CTGAN with: embedding dim = 128; generator = (256, 256); discriminator = (256, 256); learning rates = 0.0002; decay = 0.000001; batch size = 500; epochs = 300; discriminator steps = 1; pac size = 5.

TVAE: We use default parameters from https://docs.sdv.dev/sdv: class dims = (256, 256, 256, 256); random dim = 100; 64 channels; l2scale = 1e-5; batch size = 500; epochs = 300.

TabDiff Generation [shi2025tabdiffmixedtypediffusionmodel]: We use the default implementation with two changes: (i) early stopping if loss stagnates for 25 epochs, and (ii) relaxed preprocessing so that train splits are not required to retain every category in small datasets.

B Prompt Template

Listing 1: Prompt passed to Groq API

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

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

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

Output format: JSON.

Dataset name: {dataset_name}

Column names (in order): {col_names}

Summary statistics:
{summary_stats}

CSV of full data:
{data}

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.
```

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