

MALLM-GAN: Multi-Agent Large Language Model as Generative Adversarial Network for Synthesizing Tabular Data

Yaobin Ling¹, Xiaoqian Jiang¹, Yejin Kim¹

¹McWilliams School of Biomedical Informatics at UTHealth Houston

Correspondence: Yejin.Kim@uth.tmc.edu

Abstract

In the era of big data, access to abundant data is crucial to driving research forward. However, such data are often inaccessible due to privacy concerns or high costs, particularly in the healthcare domain. Generating synthetic (tabular) data can address this, but existing models typically require substantial amounts of data to train effectively, contradicting our objective of solving data scarcity. To address this challenge, we propose a novel framework to generate synthetic tabular data, powered by large language models (LLMs) that emulates the architecture of a Generative Adversarial Network (GAN). By incorporating the data generation process as contextual information and utilizing LLM as the optimizer, our approach significantly enhances the quality of synthetic data generation in common scenarios with small sample sizes. Our experimental results on public and private datasets demonstrate that our model outperforms several state-of-art models regarding generating higher quality synthetic data for downstream tasks while keeping the privacy of the real data in low data regime. Code is available at <https://github.com/yling1105/MALLM-GAN>.

1 Introduction

Tabular data is the most common data format in high-stakes sectors like healthcare. There are many fundamental problems in dealing with tabular data, such as data scarcity, missing values, and irregularity. Among them, the data scarcity problem has been the main roadblock. Many datasets in healthcare, such as clinical trial data, have small data sizes due to data collection costs and privacy risks, and consequently, these data cannot afford modern machine learning (e.g., deep learning), which generally has thousands of parameters, at a minimum.

Recent advancements in generative models, particularly in text and image, (Brown et al., 2020; Ramesh et al., 2021) have shown the benefits of

technology for generating synthetic data that resemble real data. Tabular data generation has evolved through traditional statistical approaches, such as Bayesian networks (Young et al., 2009), oversampling method (Chawla et al., 2002a), to deep learning techniques (Xu et al., 2019). However, these methods require sufficient data for training, which makes them usually overfitting and under-representative when dealing with scarce data.

Recently, advancements in large language models (LLMs) have also enabled researchers to use their general intelligence to synthesize tabular data. (Borisov et al., 2023; Hegselmann et al., 2023) The premise is that prior knowledge encoded in the parameters of LLMs can provide contextual knowledge for coherent semantics that is required to learn the underlying data generation process. Several studies transformed tabular data to natural language via serialization, and used pre-trained LLMs to generate text containing the synthetic tabular data (Borisov et al., 2023; Hegselmann et al., 2023; Li et al., 2024). However fine-tuning LLMs requires a larger sample size, contradicting the objective of addressing data scarcity. In contrast, in-context learning presents a promising alternative. In particular, a few-shot learning in in-context learning is to provide a few “examples” of data to allow LLM to learn the patterns and mimic the examples (Kaplan et al., 2020). Our study aims to utilize this few-shot capability for synthetic tabular data generation.

Therefore, our aim is to bridge this critical gap in generating synthetic tabular data with limited real data. Our key idea is to make the data generation process explicit; the objective of our in-context learning is to generate a better data generation process, as well as to generate individual data instances. Here, the data generation process is a prompt text that consists of the context of data and any simple model that describes the relationship between data variables.

	GAN	Our Model
Generator	Neural network	Frozen LLM and prompt
Discriminator	Neural network	Tabular data classifier
Optimizer	Gradient descent	Frozen LLM and prompt

Table 1: Comparison of Generative Adversarial Network (GAN) and Our Model

However, another challenge is to identify the ground-truth data generation process. Motivated by GAN’s adversarial training, we optimize the data generation process (“generator”) in adversarial training with “discriminator” (Table 1). The discriminator’s role is to discriminate real data from the generated data, and we use the accuracy of the discriminator as the loss to be minimized to optimize the generator. Unlike GAN, our generator is a text format, which does not have derivatives. We address it by prompt optimization, which leverages an independent LLM as an optimizer (Yang et al., 2024). After optimizing the data generation process, the LLM as a generator uses it to finally generate synthetic data.

The contributions of this paper can be summarized as below:

- *Novelty*: We propose a novel concept for optimizing the data generation process using in-context learning of LLM. This leverages both data-driven supervised model (discriminator) and knowledge-driven in-context learning of LLM (generator, optimizer).
- *Few-shot synthetic data generation*: Our model works when there are too few data to train a parametric model. It mitigates the data scarcity problem in healthcare studies.
- *Conditional sampling*: Our generator is based on LLM, which enables conditional sampling seamlessly by prompting.
- *Explainability*: Our LLM-based generator explicitly reveals the data generation process through prompt design. This enables transparency of our model and facilitates human feedback, such as refining the knowledge.

2 Related Studies

Synthetic tabular data generation. Synthetic data is widely used for privacy-preserving sharing and data augmentation. Classical approaches include Bayesian networks (Young et al., 2009; Upadhyaya

et al., 2023), approximate Bayesian computation (Bernton et al., 2019), and SMOTE (Chawla et al., 2002a). Bayesian networks capture pairwise causal relations via DAGs (Pearl, 2009), but struggle with nonlinear and mixed-type dependencies. Deep generative models such as VAEs (e.g., TVAE (Xu et al., 2019)), GANs (CTGAN (Xu et al., 2019)), and diffusion models (TabDDPM (Kotelnikov et al., 2023), StaSy (Song et al., 2021), Tabsyn (Zhang et al., 2024), MTabGen (Villaizán-Vallelado et al., 2025)) have become dominant. However, they require large training datasets, limiting their utility in data-scarce settings. Recently, TabPFN (Hollmann et al., 2022, 2025), a transformer-based foundation model for small tabular datasets, showed potential for data generation through meta-learning, though it remains constrained by categorical cardinality.

LLM-based synthetic data generation. LLMs excel in text generation and have been extended to tabular domains (Fang et al., 2024), including prediction (Hegselmann et al., 2023; Gulati and Roysdon, 2023; Yu et al., 2023; Li et al., 2024) and data generation (Borisov et al., 2023; Solatorio and Dupriez, 2023; Zhang et al., 2023; Gulati and Roysdon, 2023). GReaT (Borisov et al., 2023), the first such model, transformed tables into text and fine-tuned GPT-2 with column order permutations for realism. Subsequent works (e.g. Tabula (Zhao et al., 2025)) improved on this idea, but they still require expensive fine-tuning on large data. This motivates few-shot generative approaches that better address small-data regimes.

Roles of LLMs in applications. Beyond text generation, LLMs have been applied as optimizers for non-differentiable tasks, such as prompt optimization (Yang et al., 2024) or heuristic search in algorithms (Romera-Paredes et al., 2023). They also serve in multi-agent systems, where multiple LLMs collaborate on tasks like coding (Guo et al., 2024), question answering (Wu et al., 2023), and decision making (Talebirad and Nadiri, 2023; Huang et al., 2024).

LLMs and causal discovery. Causal discovery traditionally relies on conditional independence tests (Spirites et al., 2001, 1999), score-based heuristics (Tsamardinos et al., 2006a), or continuous relaxations (Zheng et al., 2018; Yu et al., 2019). Yet, recovering ground-truth structures remains difficult, especially in healthcare or data-scarce domains. Expert-driven approaches are viable but resource-intensive. Recent work (Kiciman et al., 2023) shows that LLMs, with their encoded world

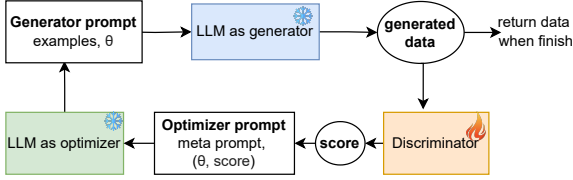


Figure 1: Overview of MALLM-GAN. In each optimization step, the **LLM as Optimizer** generates a data generation process θ in **Generator prompt** based on the pairs of previous θ and its score in **Optimizer prompt**. Then the **LLM as Generator** uses the current θ and a few examples to generate data. We evaluate the θ using the accuracy (**score**) of **Discriminator**. The more the data generation process θ is optimized, the lower the discriminator’s accuracy. This adversarial optimization finishes when data generation process is no more improved.

knowledge, can support causal reasoning and complement expert input.

In this paper, we leverage multiple LLMs with different roles to mimic adversarial training in GAN and use the heuristic causal structure discovery to guide the data generation process.

3 Methods

3.1 Problem formulation

Given a small labeled tabular dataset with n instances and d features, denoted as $D_{\text{real}} = (\mathbf{x}, y)$ where \mathbf{x} represents a d -dimensional vector of features and y indicates label. The features are described by natural-language strings like “age” or “gender”. For synthetic data generation, we train a generator on a training subset D_{train} of D_{real} , generating a synthetic dataset D_{syn} .

3.2 Multi-agent LLM as GAN

Overview. We propose to develop a multi-agent LLM as GAN (MALLM-GAN) that generates tabular data by mimicking adversarial optimization (Fig. 1). The objective is to optimize the data generation process θ , which is a natural language description of i) the problem description and ii) the simple data generation process or causal structures representing relationships between variables. In each iteration i , an LLM agent **Generator** generates data D_{syn} with θ_i and a batch in D_{train} ; a supervised model **Discriminator** is accordingly optimized using $[D_{\text{train}}, D_{\text{syn}}]$ and evaluates θ_i using D_{test} ; and another LLM agent **Optimizer** improves θ_i to decrease the discriminator’s accuracy (Algorithm 1). We repeat the iterations until the

discriminator’s accuracy converges or the iteration reaches the maximum epoch.

3.2.1 Generator

Data generation process. The data generation process θ is described in natural language and prompts the generator LLM to create synthetic data. It includes: i) context of data collection, ii) data schema, iii) causal structure describing relationships between variables, and iv) task instruction. The context provides external knowledge on data collection (e.g., “*this dataset includes subject’s socioeconomic factors...*”). The data schema contains the meta-information of variables (e.g., name, description, type, and categorical values). These elements remain constant during optimization. The causal structure, represented as a DAG and converted into text format (x_1, x_2) , indicates x_1 causes x_2 . Various serialization techniques were tested, but the original structured format proved most effective. The initial causal structure is heuristically determined (e.g., Hill climbing (Tsamardinos et al., 2006b)). The task instruction guides the goal, such as “*produce accurate and convincing synthetic data*”. Through adversarial optimization, the causal structure and instructions are refined to reduce discriminator accuracy. Thus, for each iteration i , θ_i is:

$$\theta_i = [\text{context}][\text{schema}] [\text{causal structure}]_i [\text{task instruction}]_i \quad (1)$$

Note that subscription for iteration i will be omitted for simplicity without loss of generalizability. Also, note that we used the causal structure to convey the relationship between variables within the prompt; thus, obtaining the causal structure of the ground truth is not our primary goal. (An example of generator prompt is provided in Appendix Listing 1)

Few shot examples. The data generation process θ is supplemented with n examples to leverage in-context few-shot learning. Structured data (\mathbf{x}, y) is serialized into JSON format, e.g., `{age: 53, work class: self-emp, ...}` (Detailed prompt can be found in the Appendix Section 1). Various natural language serializations were tested but had minimal impact on performance. The number n of examples is crucial; a large n allows learning from diverse examples, but may make LLM overlook instructions due to lengthy inputs; while a small n avoids overflow, but under-utilizes the data. Our solution, “batches in a batch,” splits a batch into smaller

pieces that fit the input token size, generates a set of synthetic data, and collates them into D_{syn} (see Algorithm 1 Line 6). This approach balances the trade-offs in in-context few-shot learning.

LLM as generator. The goal of the generator is to create text similar to but not identical to provided real samples, with the temperature parameter controlling the variability. The generator LLM runs multiple times with smaller examples in a batch, and the generated data is collated into D_{syn} . $D_{\text{syn},i}$ denotes the synthetic data generated at iteration i . (An example is provided in Appendix Listing 1).

3.2.2 Discriminator

Based on the generated data, we evaluate and score the quality of θ by assessing how easy it is to distinguish generated synthetic data from real data. Naturally, this is supervised learning rather than a reasoning task with LLMs. We build a discriminator f such that $f : \mathcal{X} \rightarrow c$ where $\mathbf{x} \in \mathcal{X}$ and $f(\mathbf{x})$ is the predicted label c , which is 1 if $\mathbf{x} \in D_{\text{train}}$ and 0 if $\mathbf{x} \in D_{\text{syn}}$. Specifically, at each iteration i , a new set of synthetic data $D_{\text{syn},i}$ is generated. We form the combined dataset $D_{\text{train}} \cup D_{\text{syn},i}$. We assign labels to the combined dataset by $D = \{(\mathbf{x}, c) \mid \mathbf{x} \in D_{\text{train}}, c = 1\} \cup \{(\mathbf{x}, c) \mid \mathbf{x} \in D_{\text{syn},i}, c = 0\}$. We update the discriminator f_i incrementally based on f_{i-1} . We evaluate the accuracy of the discriminator with D_{test} and pass a pair of $(\theta_i, L(f_i))$ to the optimizer where $L(f)$ denotes the discriminatory power of f (e.g., accuracy, likelihood). We prefer to use accuracy because this is a direct measurement we aim to increase and because our optimizer does not require numerical derivatives.

The discriminator obtains better discriminatory accuracy to distinguish real or synthetic data as the discriminator accumulates the discriminatory power of past iterations $0, \dots, i-1$ and is updated with newly generated, more realistic synthetic data from the current iteration i . However, on the other hand, as the D_{syn} becomes more realistic over the iterations, it gets easier to fool the discriminator, and the discriminator’s accuracy decreases. Therefore, our discriminator obtains better discriminatory power during this adversarial optimization.

3.2.3 Optimizer

The parameter to be optimized is a text, θ , which doesn’t have derivatives. So we use optimization by prompting, which leverages LLM as an optimizer (Yang et al., 2024). To make LLM act as an

optimizer, we provide a meta-prompt, which consists of two parts, the instruction and the prompt-score pairs $(\theta, L(f))$. An example is provided in Appendix Listing 3).

To leverage LLM’s in-context few-shot learning in the optimizer (Yang et al., 2024), we provide a few examples of possible solutions along with their scores from the discriminator’s scores. Note that the example here is different from data (\mathbf{x}, y) . We keep the top k solution pairs over the past iteration as the optimization trajectory to guide the optimization. We sort the score, so that the more desirable θ goes to the end of prompt. This will allow the LLM to recognize patterns among the data generation process with better score. See examples in Appendix Listing 4.

A potential pitfall is that the discriminator score $L(f)$ from past iterations $0, \dots, i-1$ is not directly comparable to the score at the current iteration i . Earlier discriminators f_0, \dots, f_{i-1} typically have weaker discriminative ability, which makes their reported scores unreliable for comparing parameter settings θ across iterations. To resolve this, we re-evaluate all past parameter candidates $\theta_0, \dots, \theta_{i-1}$ using the current discriminator f_i . In other words, instead of relying on their originally recorded scores $L(f_0), \dots, L(f_{i-1})$, we compute adjusted scores

$$\tilde{L}_i(\theta_j) = L(f_i; \theta_j), \quad j < i,$$

by passing the same candidate-generated samples through the latest discriminator. This ensures that all scores are measured against the most up-to-date discriminator, so that they are directly comparable when selecting the best θ .

In total, the LLM optimizer takes as input the meta prompt and a series of data generation process θ and adjusted scores $L(f_i)$. The optimizer outputs the revised data generation process, particularly focusing on causal structure and task instruction. We repeat iterative optimization and generation until we reach the maximum iteration.

4 Experiments

LLM backbone. We used HIPAA-compliant Azure OpenAI GPT-4o (OpenAI, 2024) as our generator and optimizer. The generator’s temperature was set to 0.5 to generate data points of highest confidence without randomly guessing, while the optimizer was set to be more creative using a temperature of 1. The top 3 prompt-score pairs

Algorithm 1: Pseudocode for MALLM-GAN’s optimization and generation

```
1: Input:  $D_{\text{train}}$  (training data),  $\theta$  (initial prompt),  
   max_epoch  
2: Output: Optimized prompt  $\theta$   
3: for  $i = 1$  to max_epoch do  
4:   for each batch in  $D_{\text{train}}$  do  
5:     Step 1: Run Generator  
6:      $D_{\text{syn}} \leftarrow [\text{generator}(\theta + \text{example}) \text{ for example in } D_{\text{train}}]$   
7:     Step 2: Run Discriminator  
8:      $c_x = 1, x \in D_{\text{real}}; c_x = 0, x \in D_{\text{syn}}$   
9:      $(D, c)_{\text{train}}, (D, c)_{\text{test}} \leftarrow \text{split}((D_{\text{real}}, c = 1)(D_{\text{syn}}, c = 0))$   
10:    Discriminator.update( $D_{\text{train}}, c_{\text{train}}$ )  
11:     $L \leftarrow \text{Accuracy}(\text{Discriminator}(D_{\text{test}}, c_{\text{test}}))$   
12:    Step 3: Run Optimizer  
13:    Pairs $_{(\theta, s)}$ .append( $(\theta, L)$ )  
14:     $\theta \leftarrow \text{Optimizer}(\text{instruction} + \text{Pairs}_{(\theta, L)})$   
15:   end for  
16: end for
```

is the to We also tried some open sourced LLM model, including Qwen3(Yang et al., 2025) and LLama3(Grattafiori et al., 2024), with different model sizes, to validate our framework’s utility with different LLM’s backbones. Open sourced models are deployed on one single NVIDIA H-100 80G.

Strong discriminators do not always contribute to a better generator (Arjovsky and Bottou, 2017). We tested Logistic regression, XGBoost, and neural network; we used the logistic regression model because it showed the highest performance while ensuring tractability during incremental updates over the iterations (Supplementary Figure 2).

Our benchmarks include several datasets from various domains: three public datasets (Adult(Becker and Kohavi, 1996), Medical Insurance(med, 2018), Asia(Scutari, 2009)), and two private medical datasets (ATACH2, ERICH) (Qureshi et al., 2016; Woo et al., 2013). To ensure fair comparison without memorization concerns of LLM (e.g., public datasets are in the training corpus of LLM), private datasets were included. Details are in Supplement Table 6.

We compare MALLM-GAN with multiple state-of-the-art tabular generative models such as traditional over-sampling techniques, SMOTE (Chawla et al., 2002b), the variational auto-encoder, TVAE (Xu et al., 2019), the generative adversarial network, CTGAN (Xu et al., 2019), the LLM-based synthetic data generation model, BeGReaT(Borisov et al., 2023), and diffusion models, TabDDPM (Kotelnikov et al., 2023) and Tabsyn (Zhang et al., 2024), the transformer-based tabular

foundation model, TabPFN,(Hollmann et al., 2025) to do generation task. Similar to MALLM-GAN, a prior work (Seedat et al., 2024) uses in-context few-shot learning of pre-trained LLMs but incorporates post-hoc data selection, which is beyond our scope. A comparison without post-hoc selection is available in Table 4. Specific hyper-parameters and computing resources are available in Supplement Section C.2.

We evaluated the impact of training data size $N = |D_{\text{train}}|$ on synthetic data quality by sampling subsets of different sizes ($N = 25, 50, 100, 200$). We particularly aimed to compare performances in low and moderate data size. For fair comparison between real and synthetic data, synthetic data was generated to match the size of real data ($|D_{\text{train}}| = |D_{\text{syn}}|$). We held out 200 samples as the test set, and replicated experiments for each subsample five times to calculate the standard error of the evaluation metrics.

5 Results

5.1 Performance Evaluation

We evaluate the performance of synthetic data generation models from two perspectives: Privacy leakage by Distance to Closest Records (DCR) and Machine Learning Efficiency (MLE) (Fang et al., 2024; Xu et al., 2019).

MLE. To evaluate the utility of our synthetic data, we train supervised models on the synthetic datasets and assess their predictive performance on real test data (D_{test}). For classification tasks (Adult, Magic, Asia), we train logistic regression, random forest, Support Vector Machine, and XGBoost classifiers, reporting the $F1$ score. For regression tasks (Insurance, ATACH, ERICH), we train linear regression, random forest, and XGBoost regressors, reporting the coefficient of determination (R^2). For each setting, we average the best scores across random seeds. As a benchmark, we also train the same models using real data (D_{train}), which serves as the gold standard maximum likelihood estimate (MLE) for comparison.

As a result, MALLM-GAN generated high-quality synthetic tabular data across multiple datasets and training data sizes, outperforming baselines (Table 2), especially with a high dimension setting ($p \gg n$, e.g. ATACH2 and ERICH). This indicates MALLM-GAN’s robustness to smaller sample sizes, unlike baselines that require more data. While TabPFN also achieves

comparable performance and scales well with increasing sample sizes, it has notable limitations—its effectiveness declines when the number of categorical levels exceeds 10 (e.g. Adult and ERICH) or when all variables are categorical (e.g. Asia), both of which are common scenarios in real-world datasets. Furthermore, MALLM-GAN outperformed the baselines in both public and private datasets, suggesting that it does not rely on the pre-trained LLM’s memorization. We also benchmark our model on the medium dataset scenario (N=400, 800). The results, as shown in Appendix Table 7, demonstrate that the data-driven model can scale their performance as the dataset size increases, while our model can still achieve a comparable performance.

DCR distributions. The DCR metric assesses the realism and diversity of synthetic data. It determines whether the synthetic data points are too similar to the real data points (potential privacy leakage) or too dissimilar (hurting the utility of the synthetic data). The DCR is defined as $d(\mathbf{x}_{\text{syn}}, D_{\text{real}}) = \min_{\mathbf{x}_{\text{real}} \in D_{\text{real}}} l_1(\mathbf{x}_{\text{syn}}, \mathbf{x}_{\text{real}})$. (Borisov et al., 2023)

Figure 2 compares DCR distributions between train and held-out sets for various models on the Adult dataset (N = 100). While baseline models such as SMOTE and TabDDPM show overfitting on the training data, MALLM-GAN achieves the comparable DCRs compared with other baselines that generated samples closely match the real distribution while maintaining diversity, reflecting low memorization risk and strong generalization. These results highlight MALLM-GAN’s capability to generate realistic and privacy-preserving data even in small-sample settings, with consistent trends observed across other datasets (Fig. 6).

5.2 Ablation study

Number n of example in in-context few shot learning. Due to the LLM’s limited context length, we implemented a "batches in a batch" method to leverage all training data within these constraints (Section 3.2.1). We varied the number n of examples and found the optimal n varied by different datasets considering their characteristics (Fig. 7). We varied the number n of few-shot examples by 1-5 and measured the MLE (Fig. 7) and DCR distribution (Table 9, 10) to find the optimal number n . We found that simply increasing number of in-context samples does not necessary improve the quality of the generation. The MLE did not in-

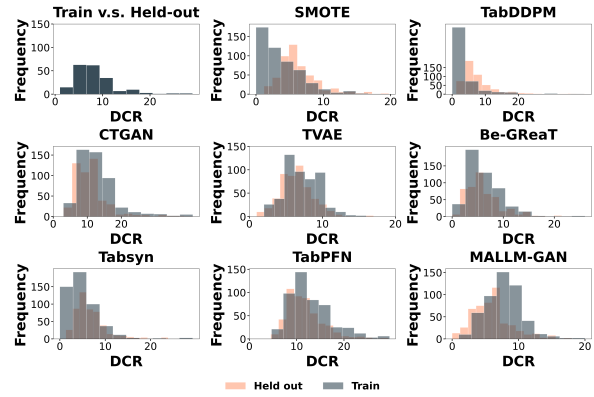


Figure 2: DCR between the synthetic data and the real data on Adult dataset. DCR was calculated based on training data and held-out test data for each model. A good model should have similar distributions between the DCR to training and the DCR to held-out dataset.

crease with more examples because more examples will increase the context length and the generator overlooks some key context information. And thus the optimal n varies among different datasets given their heterogeneous complexity and domains.

Causal structure and Optimization. To assess the impact of each component on overall performance, we examined the contribution of the causal structure in the data generation process θ and the LLM as an optimizer. We compared the full model, which includes both components, to a version without them, similar to CLLM (Seedat et al., 2024) without post-processing data selection (Table 4). It shows that incorporating the causal structure alone does not improve the MLE compared to a model with only in-context few-shot learning. However, the LLM optimizer improved θ using prior knowledge encoded in LLM and finally achieved the highest MLE. Incorporating external knowledge into LLMs has been shown to significantly improve the quality of generated text, similar to retrieval-augmented generation (RAG) (Lewis et al., 2021). Our approach shares this concept by incorporating a knowledge graph but optimizes the knowledge itself through adversarial optimization.

Experiments on Different LLM backbones We further evaluate our framework using a variety of open-source LLM backbones, and the results are presented in Table 3. As shown, the framework performs consistently well across these relatively smaller models, achieving high MLE scores. This demonstrates the generalizability of our approach and aligns with the scaling law observation that larger and more capable LLMs tend to

Table 2: Benchmark MLE results over 6 datasets. Baseline results were obtained from training the supervised models directly on the real data. SMOTE* interpolates data within the training set, thus it gets higher accuracy by copying training data and compromising DCR. "-" indicates that the model failed to generate.

		Public dataset				Private dataset	
		Adult ($F1$)	Magic($F1$)	Asia ($F1$)	Insurance(R^2)	ATACH(R^2)	ERICH(R^2)
N=25	Real data	0.80	0.79	0.83	0.52	0.25	-0.23
	SMOTE*	0.82 ± 0.02	0.58 ± 0.11	0.83 ± 0.00	0.80 ± 0.01	0.06 ± 0.20	-0.15 ± 0.12
	CTGAN	0.72 ± 0.03	0.60 ± 0.32	0.75 ± 0.10	-0.31 ± 0.29	-1.06 ± 0.70	-0.87 ± 0.97
	TVAE	0.74 ± 0.03	0.60 ± 0.04	0.84 ± 0.03	-0.02 ± 0.24	-0.01 ± 0.07	-0.13 ± 0.06
	Be-GReaT	0.76 ± 0.04	0.71 ± 0.05	0.84 ± 0.03	0.22 ± 0.20	-	-0.37 ± 0.24
	TabDDPM	0.61 ± 0.12	0.58 ± 0.11	-	-3.75 ± 0.70	-1.80 ± 1.24	-1.65 ± 0.05
	TabSyn	0.78 ± 0.04	0.74 ± 0.02	-	-0.30 ± 0.73	-	-0.49 ± 0.24
	TabPFN	0.76 ± 0.01	0.69 ± 0.04	-	0.31 ± 0.32	0.01 ± 0.29	-
	MALLM-GAN	0.82 ± 0.03	0.79 ± 0.00	0.74 ± 0.01	0.61 ± 0.12	0.17 ± 0.11	-0.20 ± 0.05
N=50	Real data	0.76	0.79	0.82	0.78	0.17	-0.01
	SMOTE*	0.74 ± 0.00	0.28 ± 0.14	0.83 ± 0.00	0.80 ± 0.01	0.20 ± 0.11	-0.10 ± 0.10
	CTGAN	0.70 ± 0.04	0.64 ± 0.10	0.41 ± 0.18	-0.48 ± 0.58	-0.53 ± 0.39	-0.00 ± 0.14
	TVAE	0.72 ± 0.02	0.70 ± 0.08	0.82 ± 0.01	0.37 ± 0.23	-0.00 ± 0.14	-0.11 ± 0.12
	Be-GReaT	0.71 ± 0.03	0.70 ± 0.03	0.80 ± 0.04	0.68 ± 0.04	-	-0.38 ± 0.12
	TabDDPM	0.59 ± 0.15	0.28 ± 0.14	-	0.63 ± 0.11	-0.88 ± 0.71	-1.02 ± 0.21
	TabSyn	0.67 ± 0.01	0.73 ± 0.03	-	0.75 ± 0.05	-	-0.44 ± 0.24
	TabPFN	0.68 ± 0.03	0.66 ± 0.05	-	0.72 ± 0.01	0.22 ± 0.07	-
	MALLM-GAN	0.74 ± 0.01	0.76 ± 0.01	0.82 ± 0.00	0.77 ± 0.03	0.18 ± 0.08	-0.07 ± 0.07
N=100	Real data	0.86	0.83	0.83	0.82	0.26	-0.04
	SMOTE*	0.78 ± 0.01	0.78 ± 0.05	0.83 ± 0.00	0.80 ± 0.01	0.27 ± 0.03	-0.15 ± 0.13
	CTGAN	0.66 ± 0.06	0.62 ± 0.03	0.63 ± 0.19	-0.09 ± 0.11	-0.40 ± 0.21	-0.33 ± 0.11
	TVAE	0.67 ± 0.05	0.70 ± 0.08	0.83 ± 0.01	0.39 ± 0.15	-0.01 ± 0.07	-0.11 ± 0.12
	Be-GReaT	0.76 ± 0.04	0.74 ± 0.02	0.83 ± 0.00	0.54 ± 0.10	-0.25 ± 0.23	-0.38 ± 0.12
	TabDDPM	0.75 ± 0.01	0.78 ± 0.05	-	-5.26 ± 0.42	-0.99 ± 0.33	-0.19 ± 0.05
	TabSyn	0.75 ± 0.02	0.74 ± 0.02	-	0.75 ± 0.02	-	-0.24 ± 0.10
	TabPFN	-	0.74 ± 0.04	-	0.71 ± 0.01	0.32 ± 0.05	-
	MALLM-GAN	0.79 ± 0.02	0.80 ± 0.01	0.74 ± 0.03	0.72 ± 0.00	0.28 ± 0.05	0.02 ± 0.05
N=200	Real data	0.85	0.81	0.83	0.83	0.27	0.16
	SMOTE*	0.78 ± 0.04	0.77 ± 0.02	0.83 ± 0.00	0.79 ± 0.02	0.31 ± 0.04	0.05 ± 0.06
	CTGAN	0.61 ± 0.02	0.59 ± 0.05	0.71 ± 0.10	-0.12 ± 0.08	-0.27 ± 0.05	-0.19 ± 0.10
	TVAE	0.67 ± 0.05	0.79 ± 0.02	0.82 ± 0.01	0.62 ± 0.05	0.08 ± 0.06	-0.08 ± 0.07
	Be-GReaT	0.69 ± 0.05	0.76 ± 0.02	0.82 ± 0.00	0.72 ± 0.03	0.16 ± 0.06	-0.18 ± 0.16
	TabDDPM	0.60 ± 0.15	0.77 ± 0.02	-	0.56 ± 0.14	-0.55 ± 0.33	-0.30 ± 0.06
	TabSyn	0.72 ± 0.08	0.79 ± 0.02	-	0.76 ± 0.04	-	-0.14 ± 0.07
	TabPFN	-	0.69 ± 0.03	-	0.79 ± 0.01	0.37 ± 0.04	-
	MALLM-GAN	0.76 ± 0.02	0.79 ± 0.01	0.83 ± 0.01	0.69 ± 0.04	0.37 ± 0.06	0.02 ± 0.02

LLM Backbone	Model Size	Adult ($F1$)	ATACH (R^2)
GPT-4o	N/A	0.82 ± 0.03	0.17 ± 0.11
Qwen3-14B	14.8B	0.82 ± 0.02	-0.13 ± 0.22
Qwen3-8B	8.19B	0.79 ± 0.01	-0.32 ± 0.10
Qwen3-4B	4.02B	0.78 ± 0.05	-1.01 ± 0.10
Qwen3-1.7B	2.03B	0.50 ± 0.09	-1.80 ± 0.31
LLama-3.1-8B-Instruct	8.03B	0.82 ± 0.03	0.07 ± 0.08
LLama-3.2-3B-Instruct	3.21B	0.58 ± 0.10	-0.23 ± 0.03

Table 3: Results with different LLM backbones on Adult dataset with N=25. Qwen3 were called without reasoning.

yield higher-quality synthetic data. Interestingly, we also observe that open-source models perform comparably to GPT-4o on public datasets but show a clear performance gap on more complex private datasets. This discrepancy may stem from potential data leakage or domain overlap in the training data of open-source models, or from the inherent limitations of smaller models when handling more complex data distributions.

Optimization trajectory of data generation process A key advantage of MALLM-GAN is its transparent, text-described data generation process, which enables direct observation of how the generation mechanism evolves during adversarial optimization. Using the Asia dataset with known causal structures of ground truth, we visualized this

Table 4: MLE of ablated models to evaluate the effects of causal structure in data generation process and optimization via LLM. Causal: Causal structure in data generation process, Opt: Optimization by LLM.

	Few-shot	Few-shot +Causal	Few-shot +Causal +Opt (MALLM-GAN)
Adult ($F1$)	0.76 ± 0.05	0.75 ± 0.04	0.79 ± 0.04
Asia ($F1$)	0.23 ± 0.00	0.28 ± 0.28	0.83 ± 0.00
Insurance (R^2)	0.68 ± 0.02	0.67 ± 0.09	0.72 ± 0.05
ATACH (R^2)	0.16 ± 0.09	0.13 ± 0.06	0.27 ± 0.07
ERICH (R^2)	-0.07 ± 0.07	0.03 ± 0.04	-0.03 ± 0.07

trajectory: the learned causal graph progressively converges to ground truth (Fig.3), as reflected by decreasing GED values. Both heuristic and uninitialized structures showed convergence, driven by knowledge from the pre-trained LLM, though with distinct patterns (Fig.4). Moreover, Table 5 discriminator accuracy declined over iterations with task instructions being increasingly specific, indicating that the synthetic data became more indistinguishable from real data.

Conditional sampling. We leverage the generator’s conditional capability to synthesize data under user-defined constraints on categorical values and numerical ranges, comparing MALLM-GAN

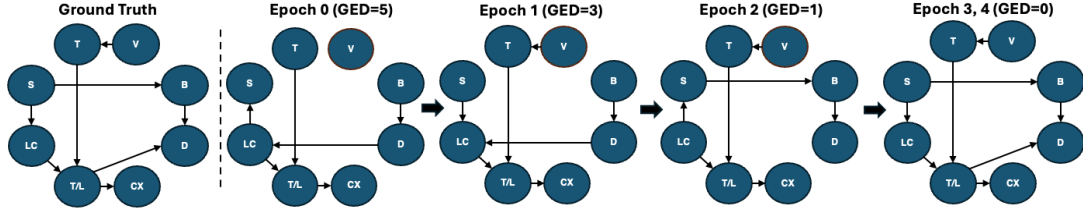


Figure 3: An example of trajectory of causal structure in data generation process over adversarial optimization using Asia dataset. T: Tuberculosis, V: Visit to Asia, S: Smoke, LC: Lung cancer, T/L: Tuberculosis or Lung cancer, CX: Chest X-ray, D: Dyspnea, B: Bronchitis. Graphical Edit Distance (GED) is used to measure the distance from the current generated causal graph to the ground truth.

Table 5: Trajectory of task instruction in data generation process over adversarial optimization. Lower score is the better.

Iteration	Task instruction	Score ↓
Epoch 1	"The ultimate goal is to produce accurate and convincing synthetic data that dutifully represents these causal relationships given the user provided samples."	100.0%
Epoch 2	"The ultimate goal is to create a detailed and convincing dataset that accurately mirrors these causal pathways. While synthesizing your data, keep in mind the following key relationships: a 'visit to Asia' increases the likelihood of 'tuberculosis', 'smoking' can lead to 'lung cancer' and 'bronchitis', and both 'tuberculosis' and 'lung cancer' can contribute to 'either tuberculosis or lung cancer', which in turn can lead to 'Dyspnea'. Also, take note of how both 'tuberculosis' and 'lung cancer' are associated with 'chest X-ray' results. Your data should reflect these intricate relationships while remaining consistent and realistic."	76.19%
Epoch 4	"You are tasked with generating a synthetic dataset that faithfully demonstrates the given causal connections. Make sure the dataset illustrates how a 'visit to Asia' can cause 'tuberculosis', how 'smoking' can lead to 'lung cancer' and 'bronchitis', and how either 'tuberculosis' or 'lung cancer' can eventually incite 'Dyspnea'. Also, the dataset should reasonably reveal how a 'chest X-ray' ties in with 'tuberculosis' and 'lung cancer'. Ensure the synthetic data reflects realistic scenarios where these factors interact, affecting each other exactly as per these defined causal relationships."	66.67%

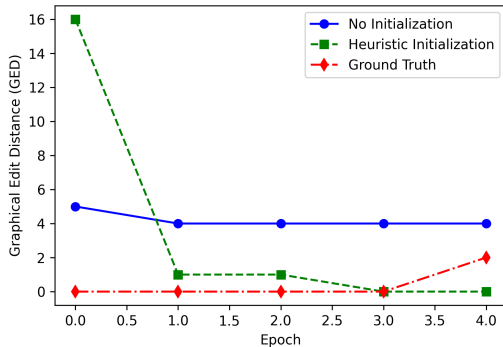


Figure 4: Causal structure initialized by different methods and the convergence step. Convergence is measured by GED to the ground truth causal graph. Experiment was conducted on the Asia dataset, which has the ground truth causal structure.

with baseline models via UMAP visualization. For categorical conditions, we selected three rare subgroups in the ERICH dataset—(i) *hematoma location = right putaminal*, (ii) *GCS score = 13*, and (iii) *prior vascular disease*—with 187, 83, and 29 patients, respectively. Baseline models failed to generate realistic samples due to limited data, whereas MALLM-GAN produced distributions closely matching the real data (Fig. 5). For the numeric condition *Age > 65* (534 patients), baselines could not model range-based constraints, but MALLM-GAN

successfully generated condition-consistent data, demonstrating flexible comprehension of natural-language conditions.



Figure 5: Real and synthetic data distribution with three categorical conditions and one numerical range condition in ERICH data.

6 Conclusions

We propose a novel framework to generate synthetic tabular data by leveraging multiple LLMs to address the data scarcity issue. Compared with other LLM-based methods, the in-context learning approach does not require fine-tuning on LLM but still leverages the whole data. We demonstrate

that LLM can help generate high-quality data for downstream tasks with an optimized prompt of domain knowledge and it enables transparent data generation with interpretation.

7 Limitations

Our proposed framework has several limitations. First, due to the limited context length of current large language models (LLMs), the method does not scale well to extremely high-dimensional datasets. As the number of features increases, the contextual input becomes excessively long, which may degrade generation quality and reduce the reliability of synthetic data. Although future LLMs with extended context capabilities may alleviate this issue, it remains a practical constraint in the current setting. Second, LLMs are known to struggle with high-quality random number generation (Hopkins et al., 2023), which can negatively affect the fidelity of synthetic data, particularly for datasets with many continuous variables where accurate stochasticity is essential. Third, similar to generative adversarial frameworks, our method lacks theoretical guarantees on convergence, which may lead to instability during training and sensitivity to hyperparameter choices. Moreover, while the proposed approach demonstrates clear advantages in low-data regimes, its relative performance gains diminish as the dataset size increases, suggesting limited benefits in large-scale settings. In addition, both training and generation incur non-trivial computational costs, especially for large datasets. Finally, although synthetic data generation is often considered a privacy-preserving alternative, it does not inherently prevent privacy leakage. In particular, adapting membership inference and attribute inference attacks to small-sample, mixed-type tabular data generated by LLM-based models remains an important open problem.

8 Ethical Considerations

Beyond the methodological limitations, we recognize several ethical risks associated with our approach. First, the optimization process in our model does not guarantee convergence, and the resulting prompts may reflect biases inherited from the pre-trained LLM backbone. This means that any optimized prompt or generated content should be interpreted cautiously, as it may be influenced by spurious correlations or latent biases in the underlying language model. Also, since our method does not

learn the true data distribution of the source domain, synthetic samples generated by the model may not be statistically representative. Consequently, these data are unsuitable for inferential analyses or for drawing causal conclusions about real-world phenomena. They should be used primarily for model benchmarking or methodological exploration rather than for policy or decision-making.

9 Acknowledgments

Dr. Yejin Kim is supported by NIH (R01AG082721, R01AG084637, U24AG098157)

References

2018. *Medical cost personal datasets*.
- Martin Arjovsky and Léon Bottou. 2017. *Towards principled methods for training generative adversarial networks*. *Preprint*, arXiv:1701.04862.
- Barry Becker and Ronny Kohavi. 1996. *Adult*. UCI Machine Learning Repository. DOI: <https://doi.org/10.24432/C5XW20>.
- Espen Bernton, Pierre E. Jacob, Mathieu Gerber, and Christian P. Robert. 2019. *Approximate Bayesian Computation with the Wasserstein Distance*. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 81(2):235–269.
- R. Bock. 2004. *MAGIC Gamma Telescope*. UCI Machine Learning Repository. DOI: <https://doi.org/10.24432/C52C8B>.
- 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*.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. *Language models are few-shot learners*. *Advances in neural information processing systems*, 33:1877–1901.
- Nitesh Chawla, Kevin Bowyer, Lawrence Hall, and W. Kegelmeyer. 2002a. *Smote: Synthetic minority over-sampling technique*. *J. Artif. Intell. Res. (JAIR)*, 16:321–357.
- Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. 2002b. *Smote: synthetic minority over-sampling technique*. *Journal of artificial intelligence research*, 16:321–357.
- Xi Fang, Weijie Xu, Fiona Anting Tan, Jiani Zhang, Ziqing Hu, Yanjun Qi, Scott Nickleach, Diego Socolinsky, Srinivasan Sengamedu, and Christos Faloutsos. 2024. *Large language models (llms) on tabular data*:

- Prediction, generation, and understanding – a survey. *Preprint*, arXiv:2402.17944.
- 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, and 542 others. 2024. [The llama 3 herd of models](#). *Preprint*, arXiv:2407.21783.
- Manbir S Gulati and Paul F Roysdon. 2023. [TabMT: Generating tabular data with masked transformers](#). In *Thirty-seventh Conference on Neural Information Processing Systems*.
- Taicheng Guo, Xiuying Chen, Yaqi Wang, Ruidi Chang, Shichao Pei, Nitesh V Chawla, Olaf Wiest, and Xi-angliang Zhang. 2024. Large language model based multi-agents: A survey of progress and challenges. *arXiv preprint arXiv:2402.01680*.
- Stefan Heggelmann, Alejandro Buendia, Hunter Lang, Monica Agrawal, Xiaoyi Jiang, and David Sontag. 2023. [Tabllm: Few-shot classification of tabular data with large language models](#). In *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of Machine Learning Research*, pages 5549–5581. PMLR.
- Noah Hollmann, Samuel Müller, Katharina Eggenberger, and Frank Hutter. 2022. [TabPFN: A transformer that solves small tabular classification problems in a second](#). In *NeurIPS 2022 First Table Representation Workshop*.
- Noah Hollmann, Samuel Müller, Lennart Purucker, Arjun Krishnakumar, Max Körfer, Shi Bin Hoo, Robin Tibor Schirrmeyer, and Frank Hutter. 2025. [Accurate predictions on small data with a tabular foundation model](#). *Nature*, 637(8045):319–326.
- Aspen K Hopkins, Alex Renda, and Michael Carbin. 2023. Can llms generate random numbers? evaluating llm sampling in controlled domains. In *ICML 2023 Workshop: Sampling and Optimization in Discrete Space*.
- Jen-tse Huang, Eric John Li, Man Ho Lam, Tian Liang, Wenxuan Wang, Youliang Yuan, Wenxiang Jiao, Xing Wang, Zhaopeng Tu, and Michael R Lyu. 2024. How far are we on the decision-making of llms? evaluating llms’ gaming ability in multi-agent environments. *arXiv preprint arXiv:2403.11807*.
- Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. [Scaling laws for neural language models](#). *Preprint*, arXiv:2001.08361.
- Emre Kıcıman, Robert Ness, Amit Sharma, and Chenhao Tan. 2023. Causal reasoning and large language models: Opening a new frontier for causality. *arXiv preprint arXiv:2305.00050*.
- Akim Kotelnikov, Dmitry Baranchuk, Ivan Rubachev, and Artem Babenko. 2023. [Tabddpm: Modelling tabular data with diffusion models](#). In *International Conference on Machine Learning*, pages 17564–17579. PMLR.
- Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2021. [Retrieval-augmented generation for knowledge-intensive nlp tasks](#). *Preprint*, arXiv:2005.11401.
- Tianhao Li, Sandesh Shetty, Advait Kamath, Ajay Jaiswal, Xiaoqian Jiang, Ying Ding, and Yejin Kim. 2024. [Cancergpt for few shot drug pair synergy prediction using large pretrained language models](#). *npj Digital Medicine*, 7(1):40.
- OpenAI. 2024. [GPT-4o System Card](#).
- Judea Pearl. 2009. *Causality*, 2 edition. Cambridge University Press, Cambridge, UK.
- Adnan I. Qureshi, Yuko Y. Palesch, William G. Barsan, Daniel F. Hanley, Chung Y. Hsu, Renee L. Martin, Claudia S. Moy, Robert Silbergleit, Thorsten Steiner, Jose I. Suarez, Kazunori Toyoda, Yongjun Wang, Haruko Yamamoto, and Byung-Woo Yoon. 2016. [Intensive blood-pressure lowering in patients with acute cerebral hemorrhage](#). *New England Journal of Medicine*, 375(11):1033–1043.
- Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. 2021. [Zero-shot text-to-image generation](#). *Preprint*, arXiv:2102.12092.
- Bernardino Romera-Paredes, Mohammadamin Barekatain, Alexander Novikov, Matej Balog, M. Kumar, Emilien Dupont, Francisco Ruiz, Jordan Ellenberg, Pengming Wang, Omar Fawzi, Pushmeet Kohli, and Alhussein Fawzi. 2023. [Mathematical discoveries from program search with large language models](#). *Nature*, 625.
- Marco Scutari. 2009. Learning bayesian networks with the bnlearn r package. *arXiv preprint arXiv:0908.3817*.
- 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 ultra low-data regimes](#). *Preprint*, arXiv:2312.12112.
- Aivin V. Solatorio and Olivier Dupriez. 2023. [Realtabformer: Generating realistic relational and tabular data using transformers](#). *Preprint*, arXiv:2302.02041.
- Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. 2021. [Score-based generative modeling through stochastic differential equations](#). In *International Conference on Learning Representations*.

- P Spirtes, C Meek, and T Richardson. 1999. An algorithm for causal inference in the presence of latent variables and selection bias (vol. 1).
- Peter Spirtes, Clark Glymour, and Richard Scheines. 2001. *Causation, prediction, and search*. MIT press.
- Yashar Talebirad and Amirhossein Nadiri. 2023. Multi-agent collaboration: Harnessing the power of intelligent llm agents. *arXiv preprint arXiv:2306.03314*.
- Ioannis Tsamardinos, Laura E Brown, and Constantin F Aliferis. 2006a. The max-min hill-climbing bayesian network structure learning algorithm. *Machine learning*, 65:31–78.
- Ioannis Tsamardinos, Laura E. Brown, and Constantin F. Aliferis. 2006b. The max-min hill-climbing bayesian network structure learning algorithm. *Machine Learning*, 65(1):31–78.
- Pulakesh Upadhyaya, Kai Zhang, Can Li, Xiaoqian Jiang, and Yejin Kim. 2023. Scalable causal structure learning: Scoping review of traditional and deep learning algorithms and new opportunities in biomedicine. *JMIR Med Inform*, 11:e38266.
- Mario Villaizán-Vallelado, Matteo Salvatori, Carlos Segura, and Ioannis Arapakis. 2025. Diffusion models for tabular data imputation and synthetic data generation. *ACM Trans. Knowl. Discov. Data*, 19(6).
- Daniel Woo, Jonathan Rosand, Chelsea Kidwell, Jacob L McCauley, Jennifer Osborne, Mark W Brown, Sandra E West, Eric W Rademacher, Salina Waddy, Jamie N Roberts, and 1 others. 2013. The ethnic/racial variations of intracerebral hemorrhage (erich) study protocol. *Stroke*, 44(10):e120–e125.
- Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Shaokun Zhang, Erkang Zhu, Beibin Li, Li Jiang, Xiaoyun Zhang, and Chi Wang. 2023. Auto-gen: Enabling next-gen llm applications via multi-agent conversation framework. *arXiv preprint arXiv:2308.08155*.
- Lei Xu, Maria Skoularidou, Alfredo Cuesta-Infante, and Kalyan Veeramachaneni. 2019. *Modeling tabular data using conditional GAN*. Curran Associates Inc., Red Hook, NY, USA.
- An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. *Qwen3 technical report*. *Preprint*, arXiv:2505.09388.
- Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V Le, Denny Zhou, and Xinyun Chen. 2024. Large language models as optimizers. In *The Twelfth International Conference on Learning Representations*.
- Jim Young, Patrick Graham, and Richard Penny. 2009. Using bayesian networks to create synthetic data. *Journal of Official Statistics*, 25(4):549–567.
- Bowen Yu, Cheng Fu, Haiyang Yu, Fei Huang, and Yongbin Li. 2023. Unified language representation for question answering over text, tables, and images. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 4756–4765, Toronto, Canada. Association for Computational Linguistics.
- Yue Yu, Jie Chen, Tian Gao, and Mo Yu. 2019. Dag-gnn: Dag structure learning with graph neural networks. *Preprint*, arXiv:1904.10098.
- 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*.
- Tianping Zhang, Shaowen Wang, Shuicheng Yan, Li Jian, and Qian Liu. 2023. Generative table pre-training empowers models for tabular prediction. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 14836–14854, Singapore. Association for Computational Linguistics.
- Zilong Zhao, Robert Birke, and Lydia Y. Chen. 2025. Tabula: Harnessing language models for tabular data synthesis. In *Advances in Knowledge Discovery and Data Mining: 29th Pacific-Asia Conference on Knowledge Discovery and Data Mining, PAKDD 2025, Sydney, NSW, Australia, June 10–13, 2025, Proceedings, Part V*, page 247–259, Berlin, Heidelberg. Springer-Verlag.
- Xun Zheng, Bryon Aragam, Pradeep Ravikumar, and Eric P. Xing. 2018. Dags with no tears: Continuous optimization for structure learning. *Preprint*, arXiv:1803.01422.

A Appendix

B Prompt Examples

Here, we provided examples of generator prompts and optimizer prompts. Note that the generator prompt evolves over the iterations.

C Experiment details

C.1 Benchmark datasets descriptions

We provide a detailed description of the benchmark data in Table 6. All the public data are licensed under CC BY-4.0. The two private datasets (ATACH2 and ERICH) are available by proper request to NIH. The private datasets have been de-identified before releasing to us for research purpose.

All the texts in the dataset, including data summary, headers, and categorical variables recorded in strings, are in English.

Listing 1: Example of generator prompt

```
1 System role:
2 % Specify role and task
3 You are a data generation model. Your task is to understand the instruction
  below and generate tabular data.
4
5 % Context of data
6 <context>The dataset include subjects social economic factors and demographics with the
  label that indicates whether their income is higher than 50k. </context><schema> age
  (numerical),workclass (categorical), education (categorical), education-num
  (numerical),marital-status (categorical), occupation (categorical), relationship
  (categorical), race (categorical), sex (categorical), capital-gain (numerical),capital-loss
  (numerical),hours-per-week (numerical),native-country (categorical), Income (categorical)
</schema><categorical variables> workclass: 'Private', 'Local-gov', 'Without-pay',
'Self-emp-not-inc', 'State-gov', 'Federal-gov', 'Self-emp-inc', education: 'Some-college',
'Masters', '11th', '1st-4th', '7th-8th', 'Bachelors', 'Doctorate', '12th', '5th-6th',
'Prof-school', 'Assoc-voc', 'Assoc-acdm', '10th', '9th', 'HS-grad', marital-status:
'Divorced', 'Married-spouse-absent', 'Married-civ-spouse', 'Never-married', 'Widowed',
'Separated', occupation: 'Handlers-cleaners', 'Transport-moving', 'Sales', 'Prof-specialty',
'Farming-fishing', 'Machine-op-inspct', 'Adm-clerical', 'Other-service', 'Craft-repair',
'Protective-serv', 'Exec-managerial', 'Tech-support', 'Priv-house-serv', relationship:
'Wife', 'Not-in-family', 'Other-relative', 'Unmarried', 'Own-child', 'Husband', race:
'Black', 'Amer-Indian-Eskimo', 'Other', 'Asian-Pac-Islander', 'White', sex: 'Male', 'Female',
native-country: 'Vietnam', 'Mexico', 'Hong', 'Taiwan', 'Italy', 'Portugal', 'Ireland',
'Guatemala', 'El-Salvador', 'United-States', Income: '>50K', '<=50K'</categorical
variables><causal structure> Consider this optimized causal graph of the data, where a pair
(A, B) is used to represent a scenario where A affects B: [(('age', 'workclass'), ('education',
'education-num'), ('education-num', 'Income'), ('marital-status', 'relationship'),
('occupation', 'Income'), ('hours-per-week', 'Income'), ('workclass', 'Income'))]This adjusted
graph introduces 'education-num', which is a key determinant of 'Income'. Be sure to reflect
'age' impact on 'workclass' and 'marital-status' effect on 'relationship'. When creating the
'Income' data, pay careful attention to the roles of 'education', 'education-num',
'occupation', and 'hours-per-week' as stated in the causal graph.</causal structure><task>
The ultimate goal is to produce accurate and convincing synthetic data that dutifully
represents these causal relationships. As such, strive for a quality score that is less than
70.0User role:<example> Here are examples from real data:['age': 53.0, 'workclass':
'Self-emp-not-inc', 'education': '10th', 'education-num': 6.0, 'marital-status':
'Married-civ-spouse', 'occupation': 'Farming-fishing', 'relationship': 'Husband', 'race':
'White', 'sex': 'Male', 'capital-gain': 0.0, 'capital-loss': 0.0, 'hours-per-week': 60.0,
'native-country': 'United-States', 'Income': '<=50K', 'age': 23.0, 'workclass': 'Private',
'education': 'HS-grad', 'education-num': 9.0, 'marital-status': 'Never-married',
'occupation': 'Adm-clerical', 'relationship': 'Own-child', 'race': 'White', 'sex': 'Female',
'capital-gain': 0.0, 'capital-loss': 0.0, 'hours-per-week': 40.0, 'native-country':
'United-States', 'Income': '<=50K']</example><instruction>Generate two synthetic samples
mimic the provided samples. DO NOT COPY the samples and try to make the generated samples
diverse. The response should be formatted strictly as a list in JSON format, suitable for
direct use in data processing scripts such as conversion to a DataFrame in Python. No
additional text or numbers should precede the JSON data.</instruction>
```

Listing 2: Example of input real example

```
1 json
2 [{"treatment": 0, "age": 68.2, "ICH volume": 4.1, "ICH Location": "L Lobar", "
  IVH volume": 0.2, "GCS score": 14.3, "NIHSS score": 11.7, "Systolic blood
  pressure": 195.0, "Diastolic Blood Pressure": 83.0, "Hypertension": 1, "
  Hyperlipidemia": 1, "Type I Diabetes": 0, "Type II Diabetes": 0, "Congestive
  heart failure": 0, "Atrial Fibrillation": 0, "PTCA": 0, "Peripheral Vascular
  Disease": 0, "Myocardial fraction": 0, "Anti-diabetic": 0, "Antihypertensives
  ": 1, "White blood count": 4.3, "Hemoglobin": 12.5, "Hematocrit": 37.7, "
  Platelet count": 129.0, "APTT": 35.3, "INR": 1.1, "Glucose": 148.0, "Sodium":
  145.0, "Potassium": 4.1, "Chloride": 106.0, "CD": 30.1, "Blood urea nitrogen
  ": 18.0, "Creatinine": 1.2, "race": "White", "sex": "Female", "ethnicity": "
  Hispanic", "mRS score after 30 days": 2.7}]
```

Listing 3: Example of optimizer prompt

```
1 System role:
2 % Specify role and task
3 Your task is to optimize prompts for generating high-quality synthetic data. Aim
  to lower the scores associated with each casual structure and prompt, where
  a lower score reflects better quality. Here are the steps:
4 1. Examine the existing prompt-score pairs.
5 2. Adjust the causal graph to better represent the underlying relationships by
  adding or removing connections, and consider incorporating new features from
  the list {self.cols}.
6 3. Modify the prompt guidance to align with the revised causal graph, ensuring
  it aids in reducing the score.
7
8 User role:
9 <pair>
10 Reflecting the adjusted causal graph of the data, where each tuple (A, B)
  indicates that A impacts B:
11 [(('age', 'workclass'), ('marital-status', 'relationship'), ('marital-status', '
  Income'), ('relationship', 'sex'), ('education', 'Income'), ('occupation', '
  Income'), ('workclass', 'Income'), ('hours-per-week', 'Income'))]
12
13 Use this causal graph as a guide to generate synthetic data that closely mirrors
  the real-world dataset. Remember to factor in the influence of 'age' on '
  workclass', and 'marital-status' on 'relationship' and 'Income'. The '
  relationship' should guide the generation of the 'sex' attribute. Further,
  take into consideration the effects of 'education', 'occupation', and 'hours-
  per-week' on 'Income' when synthesizing your data. The goal is to produce
  synthetic data that convincingly mimic these causal relationships.
14 Set your aim to achieve a score below 75.0%.
15 Score: 80.0%
16 </pair>
17
18 <pair>
19 Consider the revised and detailed causal graph of the data, which includes ('age
  ', 'workclass'), ('marital-status', 'relationship'), ('relationship', 'sex'),
  ('education', 'Income'), ('occupation', 'Income'), ('workclass', 'Income'),
  ('hours-per-week', 'Income'):
20
21 In light of the causal graph, generate synthetic samples that mimic the
  structure in the provided dataset. Values such as 'age' should reflect on '
  workclass'; 'marital-status' and 'relationship' should collaborate to inform
  'sex', while 'education', 'occupation', 'workclass', and 'hours-per-week'
  should exhibit their influence on 'Income'. Also consider 'marital-status'
  influence on 'Income'. Your aim is to generate synthetic data that fully
  embody the interconnections within this causal graph.
22 Aim to achieve a score lower than 75%
23 Score: 80.95%
24 </pair>
25
26 <pair>
27 Here is the causal graph of the data, where a tuple (A, B) indicates A causes B:
28 [(('marital-status', 'relationship'), ('marital-status', 'Income'), ('
  relationship', 'sex'))]
29 Given the description of the data, generate synthetic samples that mimic the
  provided samples.
30 Score: 85.71%
31 </pair>
32
33 Your updated prompt should explicitly include any modifications to the causal
  graph and guidance. The aim is to create a prompt that leads to the lowest
  possible score.
34
35 The updated prompt:
```

Listing 4: Example of optimizer output

```

1 <Causal structure> The optimized causal network, suggesting the influence of
  variable A on variable B, includes the following relationships: [('Age', '
  Hyperlipidemia'), ('Hyperlipidemia', 'Type II Diabetes'), ('Type II Diabetes'
  , 'Blood urea nitrogen'), ('Blood urea nitrogen', 'Creatinine'), ('
  Hypertension', 'Congestive heart failure'), ('Congestive heart failure', '
  Atrial Fibrillation'), ('Atrial Fibrillation', 'GCS score'), ('GCS score', '
  mRS score after 30 days'), ('Anti-diabetic', 'Type I Diabetes'), ('Type I
  Diabetes', 'Antihypertensives'), ('Antihypertensives', 'Potassium'), ('
  Potassium', 'Sodium'), ('PTCA', 'Peripheral Vascular Disease'), ('Peripheral
  Vascular Disease', 'Myocardial fraction'), ('Myocardial fraction', '
  Hemoglobin'), ('Hemoglobin', 'Hematocrit'), ('race', 'ethnicity'), ('Sex', '
  Hyperlipidemia')]</Causal structure>
2
3 <Task> Your task is to create realistic synthetic patient data, keeping the
  altered causal relationships as your guiding principle. Ensure the data
  reflects a diverse set of potential patient scenarios, evidencing the variety
  of health conditions one might find in a clinical setting. Remember that the
  engineered data should present unique, individual patient scenarios, each
  portraying a different, complex clinical situation. The synthetic data needs
  to be representative of different demographics ('Sex', 'race', 'ethnicity')
  and should also take into consideration different health conditions and
  treatment plans.</Task>”

```

Listing 5: Modified instruction in generator prompt for conditional sampling

```

1 <Instruction>
2 Generate {number of samples in real data meeting the conditions} synthetic
  samples with {user-provided conditions}. Response should be formatted
  strictly as a list in JSON format, suitable for direct use in data processing
  scripts such as conversion to a DataFrame in Python. No additional text or
  numbers should precede the JSON data.
3 </Instruction>

```

	# samples	# features	Description	Source
Adult	32,561	14	The dataset includes people’s socioeconomic factors and demographics, with the label that indicates whether their income is higher than 50k.	(Becker and Kohavi, 1996)
Magic	19,020	10	It is a simulated registration of high-energy gamma particles in a ground-based atmospheric Cherenkov gamma telescope using the imaging technique	(Bock, 2004)
Medical Insurance	2,772	7	This dataset describes the patients’ demographics with their health insurance bills.	(med, 2018)
Asia	10000	8	This is the dataset used to illustrate the utility of Bayesian network to do causal structure discovery. The dataset is available in the R-package.	(Scutari, 2009)
ATACH2	1,000	37	This is an RCT data that investigate in treatment for Intracerebral hemorrhage patients.	(Qureshi et al., 2016)
ERICH	1,521	29	The data is from a case-control study of Intracerebral Hemorrhage study which aims to investigate in the Ethnic/Racial variations.	(Woo et al., 2013)

Table 6: Datasets description

C.2 Hyperparameters

Specific hyperparameters for each model are provided below.

- **CTGAN**: Default parameters
- **TVAE**: Default parameters
- **BeGReaT**:
 - Base LLM: Distilled-GPT2
 - Batch size: 40
 - Epochs: Depend on the feature numbers and the total sample size. (200-400)
- **MALLM-GAN**:
 - Temperature for generator: 0.5
 - Temperature for optimizer: 1.0
 - Batch size: 50
 - Discriminator: XGBoost (max depth: 3, eta: 0.3, objective: binary:logistic)
- **TabDDPM**: Default parameters
- **Tabsyn**: Default parameters

D Additional Experiments on Medium Size Datasets

To evaluate our model’s performance scaling with larger sample sizes, We also benchmark our model on the datasets of medium sample sizes(N=400, 800). The results are shown in Table 7.

D.1 DCR evaluation on other datasets

The following figures are to evaluate DCR on other datasets:

E Comparison of different discriminators

In the study, we compare 3 different kinds of supervised classification models as the role of a discriminator. An experiment was conducted on the Adult dataset’s sub-sample to demonstrate the discriminator’s effects on the quality of the generated data.

F Computing resource details

The model proposed in this study does not require extensive computing resources for fine-tuning. However, this model requires access to the Azure service. For other baseline models, they are implemented on an NVIDIA A100 40GB GPU.

G Ablations studies on the number of provided real samples

We conducted experiments to test the number of real examples’ effects on the downstream evaluation metrics. Table 9 and Table 10 shows the DCR distance to the training and testing datasets respectively. We can learn from the tables that there is no association between the number of examples and the DCR.

Also, the Figure 7 shows the MLE efficacy given different in-context number of shots. We can see that the patterns differentiate among different datasets. It is because that the complexity of the data can affect the context length and thus cast effect on the generation quality of the data.

H Cost Analysis

The section demonstrates some examples of the time cost of our framework on the real world datasets. We provide both the training time and testing time in Table 11.

This is an appendix.

		Adult ($F1$)	Public dataset Magic($F1$)	Asia ($F1$)	Private dataset		
					Insurance(R^2)	ATACH(R^2)	ERICH(R^2)
N=400	Real data	0.83	0.82	0.84	0.85	0.31	0.18
	SMOTE*	0.85 ± 0.03	0.81 ± 0.01	0.84 ± 0.00	0.83 ± 0.00	0.32 ± 0.02	0.07 ± 0.05
	TabDDPM	0.82 ± 0.03	0.81 ± 0.01	-	0.79 ± 0.03	0.36 ± 0.02	0.09 ± 0.04
	CTGAN	0.63 ± 0.02	0.60 ± 0.02	0.59 ± 0.17	-0.18 ± 0.10	-0.08 ± 0.07	-0.24 ± 0.10
	TVAE	0.71 ± 0.07	0.79 ± 0.01	0.71 ± 0.07	0.62 ± 0.05	0.16 ± 0.08	-0.19 ± 0.06
	Be-GReaT	0.79 ± 0.04	0.79 ± 0.02	0.79 ± 0.00	0.72 ± 0.03	0.20 ± 0.06	-0.13 ± 0.07
	TabSyn	0.83 ± 0.02	0.80 ± 0.01	-	0.82 ± 0.02	0.40 ± 0.04	-
	TabPFN	-	0.77 ± 0.02	-	0.71 ± 0.01	0.40 ± 0.04	-
MALLM-GAN	0.79 ± 0.02	0.80 ± 0.01	0.83 ± 0.00	0.71 ± 0.03	0.27 ± 0.04	0.02 ± 0.03	
N=800	Real data	0.71	0.81	0.84	0.85	0.40	0.21
	SMOTE*	0.71 ± 0.03	0.82 ± 0.01	0.84 ± 0.00	0.83 ± 0.00	0.37 ± 0.03	0.10 ± 0.05
	TabDDPM	0.70 ± 0.03	0.82 ± 0.01	-	0.83 ± 0.01	-0.53 ± 0.45	0.12 ± 0.04
	CTGAN	0.64 ± 0.05	0.54 ± 0.05	0.48 ± 0.06	-0.41 ± 0.06	-0.05 ± 0.06	-0.04 ± 0.02
	TVAE	0.77 ± 0.02	0.78 ± 0.01	0.82 ± 0.01	0.68 ± 0.01	0.12 ± 0.07	-0.05 ± 0.03
	Be-GReaT	0.75 ± 0.07	0.75 ± 0.01	0.82 ± 0.00	0.53 ± 0.21	0.00 ± 0.07	-0.04 ± 0.05
	TabSyn	0.78 ± 0.08	0.82 ± 0.01	-	0.82 ± 0.01	0.42 ± 0.04	-
	TabPFN	-	-	-	0.71 ± 0.01	0.40 ± 0.04	-
MALLM-GAN	0.80 ± 0.02	0.79 ± 0.00	0.84 ± 0.00	0.72 ± 0.01	0.36 ± 0.02	0.02 ± 0.02	

Table 7: Experiments on sample size 400, 800

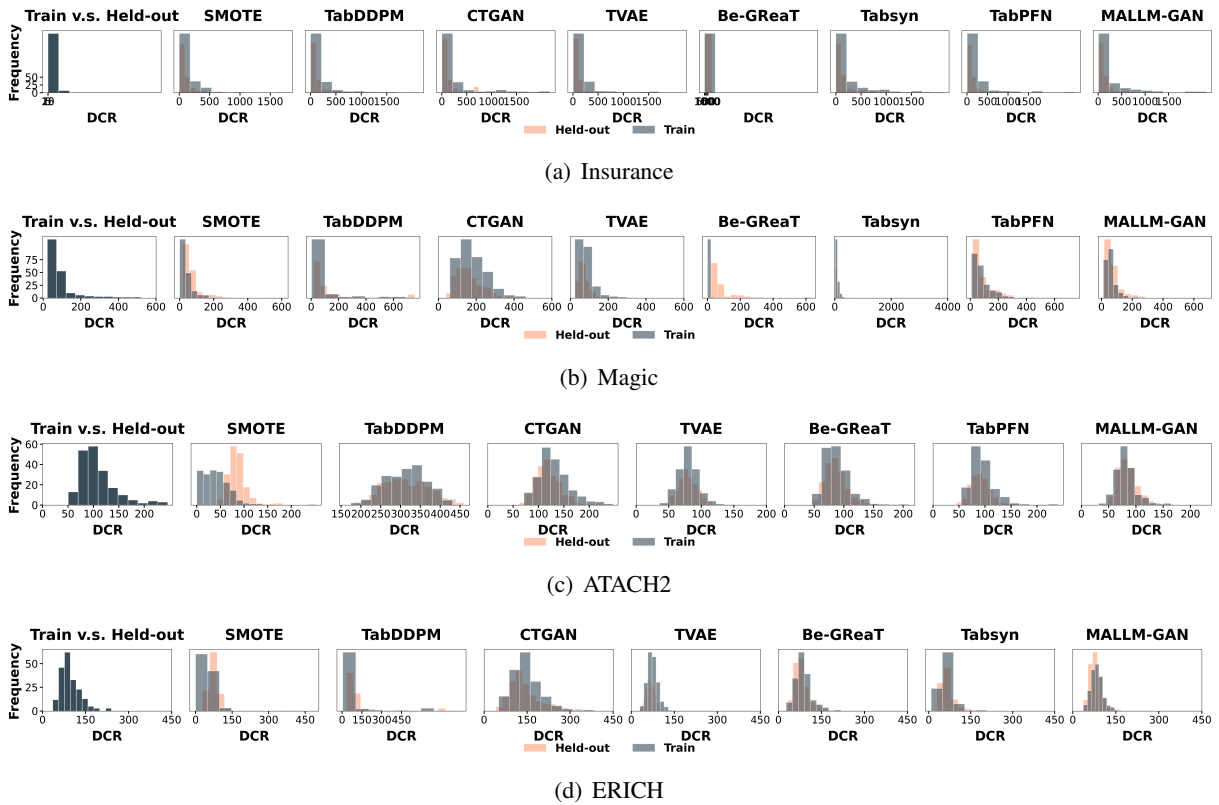


Figure 6: DCR evaluation results.

	N = 100	N = 200	N = 400	N = 800
XGBoost	0.78 ± 0.03	0.73 ± 0.01	0.76 ± 0.06	0.72 ± 0.00
Logistic regression	0.79 ± 0.02	0.77 ± 0.02	0.79 ± 0.03	0.80 ± 0.02
Neural Network	0.80 ± 0.02	0.57 ± 0.12	0.78 ± 0.06	0.67 ± 0.12

Table 8: Comparison of different discriminators' effects on the quality of the synthetic data. An experiment on sub-sample of Adult data.

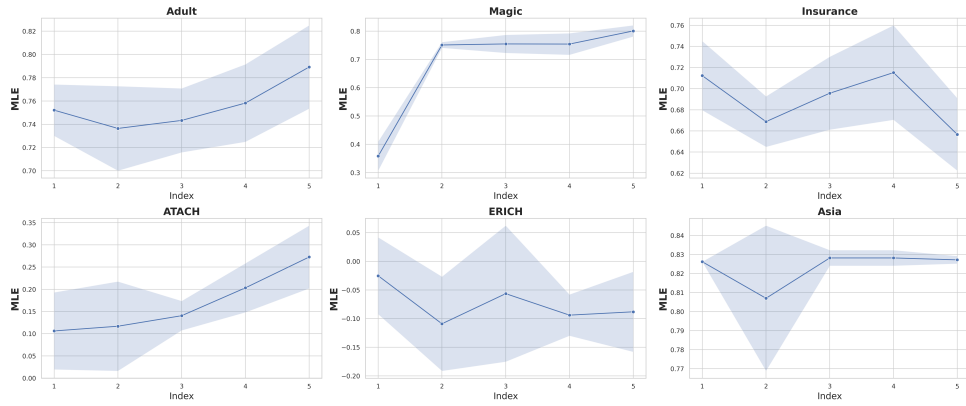


Figure 7: Number n of examples and MLE.

	1	2	3	4	5
Adult	5, 6, 10	5, 7, 12	4, 6, 9	4, 6, 10	4, 6, 11
Magic	23, 29, 37	24, 33, 52	30, 44, 65	37, 53, 72	40, 55, 73
Insurance	31, 93, 301	44, 66, 453	32,60, 182	33, 73, 167	29, 55, 168
ATACH2	61, 73, 88	72, 87, 97	69, 78, 89	66, 75, 94	67, 83, 104
ERICH	53, 61, 79	59, 78, 96	59, 74, 101	56, 72, 96	50, 64, 88

Table 9: Number n of examples and DCR to **training dataset**. 25%, 50% (Median), 75% quantile.

	1	2	3	4	5
Adult	4, 7, 10	5, 7, 11	5, 6, 10	4, 7, 10	4, 7, 11
Insurance	30, 115, 337	34, 91, 405	36, 76, 245	24, 64, 170	27, 70, 150
Magic	45, 61, 87	44, 57, 82	45, 58, 88	46, 59, 86	45, 60, 86
ATACH2	84, 100, 120	82, 99, 122	81, 97, 125	79, 98, 124	82, 103, 128
ERICH	70, 87, 110	66, 82, 111	51, 82, 104	62, 80, 108	62, 80, 117

Table 10: Number n of examples and DCR to **held out dataset**. 25%, 50% (Median), 75% quantile.

Dataset	Sample Size	Number of Epochs	Number of Examples Per Call	Training Time	Inference Time
Asia	50	10	4	13.5	5.9
Asia	200	4	4	24.8	3.8
ERICH	100	5	1	62.7	14.6
ERICH	200	4	1	92.5	23.7

Table 11: Examples of computational cost analysis. (Unit: Minutes)