Self-Reinforcing Controllable Synthesis of Rare Relational Data via Bayesian Calibration

Chongsheng Zhang[*†1,2] **Hao Wang**[*1] **Zelong Yu**[*1] **Esteban Garces Arias**[*2,3]
Julian Rodemann[*2,4] **Zhanshuo Zhang**[1] **Qilong Li**[1] **Gaojuan Fan**[1]
Krikamol Muandet[4] **Christian Heumann**[2]

[1]Henan University, China [2]Department of Statistics, LMU Munich [3]MCML Munich
[4]CISPA Helmholtz Center for Information Security, Saarbrücken, Germany

Correspondence: cszhang@henu.edu.cn

Abstract

Imbalanced data are commonly present in real-world applications. While data synthesis can effectively mitigate data scarcity for rare classes, and LLMs have revolutionized text generation, the application of LLMs to the synthesis of relational/structured tabular data remains underexplored. Moreover, existing approaches lack an effective feedback mechanism to guide LLMs in continuously optimizing the quality of the generated data throughout the synthesis process. In this work, we propose RDDG, **R**elational **D**ata generator with **D**ynamic **G**uidance, which is a unified in-context learning framework that employs progressive chain-of-thought (CoT) steps to generate tabular data for enhancing downstream imbalanced classification performance. RDDG first uses core set selection to identify representative samples from the original data, then utilizes in-context learning to discover the inherent patterns and correlations among attributes within the core set, and subsequently generates tabular data while preserving the aforementioned constraints. More importantly, it incorporates a self-reinforcing feedback mechanism that provides automatic assessments of the quality of the generated data, enabling continuous quality optimization throughout the generation process. Experimental results on multiple real and synthetic datasets demonstrate that RDDG outperforms existing approaches in both data fidelity and downstream imbalanced classification performance. We make our code available at https://github.com/cszhangLMU/RDDG.

1 Introduction

The scarcity of data, particularly annotated data, has been a pervasive challenge in deep learning and the new era of artificial intelligence (AI). Automated data synthesis techniques are now widely used to address data scarcity, particularly with the rise of large foundation models. For instance, large language models (LLMs) have been widely used in various text generation and analysis tasks (Zhang et al., 2023a; Liang et al., 2024), while multi-modal large foundation models have also been used to generate image data to enhance visual learning performance (Zhao et al., 2024a).

Despite the revolutionary impact of foundation models and their widespread applications in *unstructured* text and image generation, their potential for relational or structured tabular data synthesis is still underexplored (Kim et al., 2024), whereas non-LLM-based approaches have already demonstrated their effectiveness in relational data generation for enhancing imbalanced classification performance (Yoon et al., 2020; Ishfaq et al., 2018; Kotelnikov et al., 2022). Moreover, there is no internal self-reinforcing feedback mechanism to guide foundation models to continuously optimize the quality of generated data throughout the in-context data synthesis process.

To address the above challenges, we propose RDDG, a novel framework with progressive chain-of-thought (CoT) steps and dynamic guidance for in-context synthesis of tabular data using LLMs. RDDG first performs core set selection based on sample error variance to identify the most representative training samples, which are then fed into LLMs along with initial prompts that contain dataset and attribute information, thereby guiding the LLMs to analyze functional relationships among attributes. With these patterns and constraints in hand, RDDG continually feeds each batch of real data to LLMs, guiding them to generate semantically meaningful samples. This procedure incorporates a self-reinforcing feedback mechanism that automatically assesses the quality of the generated data relative to real data. The evaluation results are then converted into feedback prompts to be integrated into the subsequent in-context learning, enabling continuous optimization of data qual-

[*] Equal contribution
[†] Corresponding author

Findings of the Association for Computational Linguistics: ACL 2026, pages 8690–8711
July 2-7, 2026

ity throughout the synthesis process.

Contributions The main contributions of this paper are as follows:

- We propose a unified in-context learning framework with progressive CoT steps for generating tabular data to enhance imbalanced classification, which obtains domain-specific prior knowledge by mining functional relationships, leading to constraint-driven, controllable synthesis of tabular data.
- We devise a self-reinforcing feedback mechanism that uses multiple quality measures to automatically assess the quality of the current batch of generated data, which is then converted into feedback prompts and propagated to the following in-context learning process, thereby continuously improving the realism of the generated data throughout the synthesis process.
- We formulate the self-reinforcing feedback process as a Bayesian calibration problem, prove Bayes-optimality of our framework (Theorem 1), and show that our feedback mechanism targets these optimal strategies under some assumptions (Proposition 1).
- We conduct extensive experiments on eight datasets. Compared to the state-of-the-art in-context learning approaches for imbalanced classification, RDDG achieves average improvements of more than 2% and 1% in the weighted Macro-F1 metric and Balanced Accuracy, respectively, while simultaneously preserving superior data fidelity.

2 Related Work

Based on neural architectures, deep data generation methods can be divided into statistical distribution-based methods (Liu et al., 2020; Wang et al., 2022), GAN-based methods (Xu et al., 2019; Park et al., 2018; Wang et al., 2020; Zhang et al., 2023b; Zhao et al., 2024b), diffusion model-based methods (Kotelnikov et al., 2022; Mueller et al., 2025), and foundation model-based methods (van Breugel and van der Schaar, 2024; Hollmann et al., 2025, 2023; Borisov et al., 2022; Kim et al., 2024).

Statistical distribution-based methods focus on modeling underlying data distributions and generate samples via distribution sampling, whereas GAN-based methods utilize the GAN architecture for data generation. TableGAN (Park et al., 2018) pioneered GAN-based tabular data synthesis by incorporating an additional classifier module to co-supervise the generation process. GL-GAN (Wang et al., 2020) first adopts SMOTE (Chawla et al., 2002) to create minority samples, then utilizes GANs to learn underlying data distributions. CTAB-GAN (Zhang et al., 2023b) and CTAB-GAN+ (Zhao et al., 2024b) jointly optimize adversarial and classification losses to improve minority class sample generation capabilities.

TabDDPM (Kotelnikov et al., 2022) investigates Diffusion model-based tabular data synthesis, generating high-quality samples through forward noise injection and reverse denoising processes. CDTD (Mueller et al., 2025) adapts the diffusion model for generating mixed-type features, adaptively scaling the respective losses for numeric and categorical features.

The emergence of pretrained foundation models has opened new avenues for data synthesis (van Breugel and van der Schaar, 2024), and existing approaches in this direction can be further categorized into fine-tuning and prompt-based methods. TabPFN (Hollmann et al., 2025, 2023) is a Transformer-based tabular foundation model that modifies the self-attention mechanism so that training samples can only attend to other training samples, pre-trained on 100 million synthetic datasets generated using structural causal models. Unlike TabPFN, TABULA-8B (Gardner et al., 2024) builds a tabular foundation model by fine-tuning Llama 3-8B using a filtered 4-million high-quality subset of real-world web-crawled tables, converting relational records into template-based prompts. Similarly, LLM-GTL (Wen et al., 2024) converts tabular samples into template-based, instruction-oriented prompts and fine-tunes Llama-2 using 340 real-world Kaggle datasets. GReaT (Borisov et al., 2022) and EPIC (Kim et al., 2024) employ prompt-based methods that leverage in-context learning to guide LLMs in generating synthetic tabular data without requiring model updates or fine-tuning.

Discussion. Despite the above advances in LLM-based tabular data generation, i) a significant gap exists between data generation methods and downstream task optimization goals, particularly imbalanced classification; and ii) there is a lack of an internal feedback mechanism that can guide LLMs to continuously optimize the quality of the generated data throughout the in-context learning process.

3 Methodology

Given a sample of real data $\mathcal{R}$ drawn from some unknown distribution $\mathbb{P}_{\mathcal{R}}$, our goal is to generate synthetic data $\mathcal{S}$ via $\mathcal{S} = S_\phi(\mathcal{R}, \mathcal{C}, \mathcal{F})$ where $S_\phi(\cdot,\cdot,\cdot)$ is the generator parametrized by ϕ, $\mathcal{C}$ is a set of constraints, and $\mathcal{F}$ is the feedback. The feedback $\mathcal{F}$ constitutes the *self-reinforcing* synthesis mechanism, while the constraints $\mathcal{C}$ render the method *controllable*.

Our procedure is sequential, i.e., for $i \in \mathcal{I}$, batches of synthetic data $\mathcal{S}_i$ are generated by

$$\mathcal{S}_i = S_\phi(\mathcal{R}_i, \mathcal{C}, \mathcal{F}_{i-1}) \tag{1}$$

with $\mathcal{I}$ being an index set, typically a finite subset of natural numbers $\mathbb{N}$. Notably, the sequential setup invokes the Bayesian perspective on simulation (Wade et al., 2022), as detailed in Section 3.5.

Objective At any iteration $i \in \mathcal{I}$, we aim to generate synthetic data $\mathcal{S}_i$ that is close to the unknown distribution $\mathbb{P}_{\mathcal{R}}$ with respect to some divergence measure d (e.g., Kullback-Leibler (KL) divergence), i.e.,

$$\min_{\mathcal{S}_i} \; d(\hat{\mathbb{P}}_{\mathcal{S}_i}, \mathbb{P}_{\mathcal{R}}) \tag{2}$$

with $\hat{\mathbb{P}}_{\mathcal{S}_i}$ being the empirical measure of $\mathcal{S}_i$ for any $i \in \mathcal{I}$. In the following, we present our method that aims at achieving this goal by approximating $\arg\min_{\mathcal{S}_i} d(\hat{\mathbb{P}}_{\mathcal{S}_i}, \mathbb{P}_{\mathcal{R}})$, including theoretical intuition based on a Bayesian perspective in Section 3.5.

3.1 Overall Framework

In Fig. 1, we present our RDDG framework, i.e., **R**elational **D**ata generator with **D**ynamic **G**uidance, which is a progressive in-context learning framework with an internal self-reinforcing feedback mechanism to produce constraint-compliant, high-fidelity relational data for enhancing downstream imbalanced classification performance.

Prior to in-context learning, given the context-window limitation of LLMs, we first curate a representative core set from the training data to ensure comprehensive coverage of the data's characteristics.

The second step is relation mining, in which we use in-context learning to uncover latent patterns and inter-attribute relationships within the core set, which are then established as explicit structural constraints to guide LLMs in the generation process.

The third step is data generation and constraint optimization, in which RDDG incorporates an internal self-reinforcing feedback mechanism that automatically assesses the quality of the generated data from the preceding batch, with the corresponding evaluation results being converted into feedback prompts and incorporated with the explicit constraints into subsequent in-context learning, to guide LLMs to continuously improve the quality/fidelity of the generated data. This batch-wise generation paradigm continues until the total number of synthetic samples reaches the target threshold. A detailed overview of these steps is provided as a pseudo-algorithm in Appendix A.7.

3.2 Core Set Construction

To provide high-quality training samples to LLMs under prompt-length constraints, we propose grafting the core set sample selection method (Hong et al., 2024), originally designed for processing high-dimensional medical data in resource-limited environments, to address the context-length limitation of LLMs.

In our design, we use a straightforward MLP model and split the overall training process into early, middle, and late stages. As with Hong et al. (2024), we aim to identify samples that display high variance in prediction error across both early and late training stages. Specifically, we calculate the prediction error (L2 error) of the MLP model on each tabular sample at every epoch, denoted in Equation 3, then calculate the mean and variance of each sample's prediction errors across both the early and late training stages/epochs. Finally, for each class, we select the Top-K samples that exhibit the highest variance (*Var*) in prediction errors, as depicted in Equation 4. If the number of samples in a class is less than K, we repeat sampling for that class.

$$\mathcal{L}_2(\mathbf{y}_{\text{pred}}, \mathbf{y}_{\text{true}}) = \|\mathbf{y}_{\text{pred}} - \mathbf{y}_{\text{true}}\|_2^2 \tag{3}$$

$$\text{Top}_k(k) = \arg top_K([\text{Var}_i \mid i \in N_k]) \tag{4}$$

Overall, our class-balanced core set selection strategy identifies, for each class, the Top-K samples with the highest variance in prediction errors, ensuring that each class receives fair attention when fed into LLMs. More details are given in Appendix A.4.

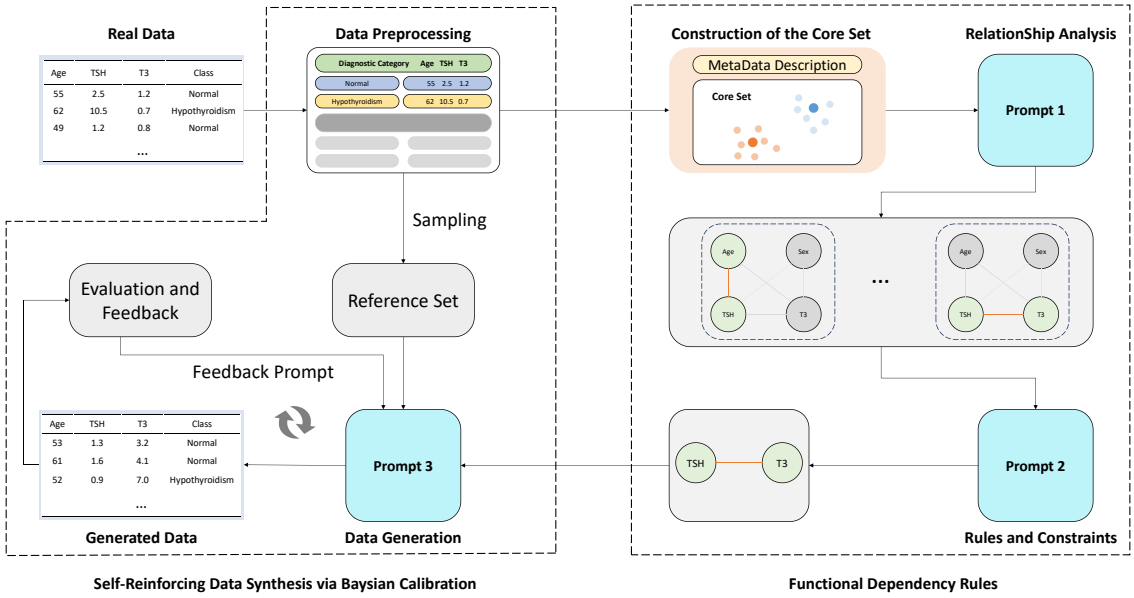


Figure 1: Overall framework of RDDG, consisting of three main steps, which are core set construction, relation mining, and data generation and constraint optimization. (i) Core set construction selects representative samples from original data to address the context-length limitation of LLMs; (ii) Relation mining uncovers the latent functional relationships (patterns and inter-attribute correlations) from core set (prompts 1 and 2); (iii) Batch-wise data generation considers both the functional relationships from step (ii) and a batch of reference set (prompt 3_1), and devises a self-reinforcing feedback mechanism (prompt 3_2) for continuously optimizing in-context learning.

3.3 Progressive In-Context Data Synthesis

Relation Mining. Our RDDG framework resembles the "chain-of-thought (CoT)" process (Wei et al., 2022), with step-by-step reasoning steps to guide LLMs in the generation process. To generate realistic synthetic data, we guide LLMs to first analyze the functional relationships (mainly the patterns and inter-attribute relationships) from the core set (prompt 1); next, we instruct LLMs to comprehensively take the core set, metadata, and the above relationships into account to establish explicit rules and constraints for data generation (prompt 2).

Data Generation. In the data generation phase, the original training set is partitioned into multiple subsets (batches) and used as reference sets for in-context learning. In each iteration, given a reference set and the above rules and constraints from relation mining, we guide LLMs to generate synthetic samples (prompt 3_1).

To further control the quality of the generated data, we introduce the self-reinforcing feedback mechanism for constraint optimization below.

3.4 Dynamic Guidance Adjustment

To ensure continuous improvement in the quality of generated data across subsequent batches, we integrate a novel self-reinforcing feedback mechanism into the synthesis process, which provides automatic evaluations of the current batch's quality to dynamically guide in-context learning.

Specifically, for each batch i of real data (reference set), we generate a corresponding batch of synthetic data, then immediately evaluate its quality through three key dimensions: Statistical Consistency, which involves comparing means and standard deviations between generated and real data; Correlation Preservation, utilizing Pearson correlation coefficients to assess the maintenance of inter-attribute relationships; and Distribution Consistency, employing Kolmogorov-Smirnov tests to verify distributional alignment.

Crucially, the feedback from evaluating batch i is not used to regenerate the same batch, but rather incorporated as additional guidance when processing batch $i+1$ with a new subset of real data. Formally, let $\mathcal{R}_i$ denote the i-th batch of real data (reference set) and $\mathcal{S}_i$ the corresponding synthetic batch. The generation process follows:

$$\mathcal{S}_i = S_\phi(\mathcal{R}_i, \mathcal{C}, \mathcal{F}_{i-1}) \tag{5}$$

where S_ϕ is the generation mechanism (treated as a random variable) with hyperparameter ϕ, $\mathcal{C}$ represents the constraints derived from relation mining, and $\mathcal{F}_{i-1}$ represents the feedback guidance computed from evaluating the previous batch $\mathcal{S}_{i-1}$ against $\mathcal{R}_{i-1}$.

This forward-propagating feedback mechanism creates a self-optimizing generation pipeline where each iteration benefits from insights gained in previous rounds/batches. The quality evaluation results are then transformed into a prompt (prompt 3_2) and incorporated into the existing prompt (prompt 3_1) for the next batch generation. This sequential refinement continues until the total number of synthetic samples reaches the target threshold, progressively enhancing the semantic consistency and statistical fidelity of synthesized data. Detailed prompt designs are provided in Tables 12, 13, and 14 in the Appendix A.5.

3.5 A Bayesian View on Dynamic Guidance Adjustment

Our RDDG framework with a self-reinforcing feedback mechanism can be understood from a Bayesian perspective on simulation (Wade et al., 2022; Poole and Raftery, 2000; Andradóttir and Bier, 2000). Recall from Section 3.4 that we generate synthetic batches $\mathcal{S}_i$ via $\mathcal{S}_i = S_\phi(\mathcal{R}_i, \mathcal{C}, \mathcal{F}_{i-1})$ with hyperparameters ϕ, where $\mathcal{R}_i$ is real data, $\mathcal{C}$ are constraints, and $\mathcal{F}_{i-1}$ being the feedback guidance from previous $\mathcal{S}_{i-1}$ against $\mathcal{R}_{i-1}$. Bayesian calibration (Wade et al., 2022) treats ϕ as unknown and places a prior $p(\phi)$ encoding structural beliefs discovered during the relation mining phase.

Given $\mathcal{R}_i$, S_ϕ and summary targets $T(\mathcal{R})$ (means, standard deviations, Pearson correlations, KS distances), we posit a likelihood $p(T(\mathcal{R}) \mid T(S_\phi))$ that scores synthetic batches against $\mathcal{R}$. Calibration then infers the posterior $p(\phi \mid T(\mathcal{R})) \propto p(T(\mathcal{R}) \mid T(S_\phi))\, p(\phi)$. The *closed loop* appears as sequential Bayesian updating over batches $i = 1, 2, \ldots$, where feedback metrics F_i act as posterior predictive checks; each update nudges ϕ toward regions that simultaneously preserve functional relations and improve class-imbalance targets, thereby shrinking the discrepancy while quantifying uncertainty in ϕ and induced predictions. This mirrors standard Bayesian calibration steps (Wade et al., 2022) and enables casting Bayes-optimal prompting as posterior expected-utility maximization.

Theorem 1 (Bayes-optimal prompting) *Let $\phi \in \Phi$ (compact) denote the prompt/control of the synthesizer S_ϕ, $T(\mathcal{R})$ be the target summaries from the real data, and the Bayesian calibration posterior for ϕ be*

$$\pi(\phi \mid T(\mathcal{R})) \;\propto\; p(T(\mathcal{R}) \mid T(S_\phi))\; p(\phi).$$

Fix a bounded, jointly upper semicontinuous utility $U(\phi, \phi'; T(\mathcal{R}))$ in $(\phi, \phi') \in \Phi \times \Phi$, measuring the performance of action ϕ when the true synthesizer parameter is ϕ'.[1] Then the Bayes-optimal prompt

$$\phi^\star \;\in\; \arg\max_{\phi \in \Phi} \mathbb{E}_{\phi' \sim \pi(\cdot \mid T(\mathcal{R}))}\big[\, U(\phi,\, \phi'; T(\mathcal{R}))\,\big]$$

exists (by compactness of Φ and upper semicontinuity of U) and minimizes posterior expected regret against every admissible $\tilde{\phi} \in \Phi$:

$$\mathbb{E}_{\phi' \sim \pi(\cdot \mid T(\mathcal{R}))}\big[\, U(\tilde{\phi},\, \phi'; T(\mathcal{R})) - U(\phi^\star,\, \phi'; T(\mathcal{R}))\,\big] \;\le\; 0.$$

Proof 1 (Sketch) *Define the posterior expected utility functional $\mathcal{V}(\phi) := \mathbb{E}_{\phi' \sim \pi(\cdot \mid T(\mathcal{R}))}[U(\phi, \phi'; T(\mathcal{R}))]$. Since U is bounded and jointly upper semicontinuous, $\mathcal{V}(\phi)$ is upper semicontinuous in ϕ. Compactness of Φ then guarantees existence of $\phi^\star \in \arg\max_{\phi \in \Phi} \mathcal{V}(\phi)$ by the extreme value theorem. By standard Bayesian decision theory (Berger, 2013), maximising $\mathcal{V}$ is precisely the Bayes rule: for every $\tilde{\phi} \in \Phi$,*

$$\mathcal{V}(\phi^\star) \;\ge\; \mathcal{V}(\tilde{\phi}),$$

which is equivalent to the stated regret inequality. Under strict concavity of $\mathcal{V}$ and convexity of Φ, $\phi^\star$ is unique. ■

This standard result from Bayesian decision theory implies that the self-reinforcing feedback mechanism can target the Bayes-optimal prompt, which is central to the following proposition.

Proposition 1 *Let $\mathcal{V}(\phi) := \mathbb{E}_{\phi' \sim \pi(\cdot \mid T(\mathcal{R}))}\big[U(\phi, \phi'; T(\mathcal{R}))\big]$ denote the posterior expected utility, with U as in Theorem 1 depending on* both *the action ϕ and the uncertain parameter ϕ'. Suppose the self-reinforcing synthesis loop generates batches $\mathcal{S}_i$ yielding a stochastic supergradient g_i satisfying:*

[1] For instance, $U(\phi, \phi'; T(\mathcal{R}))$ could trade off fidelity to $T(\mathcal{R})$ and task utility, e.g., $U(\phi, \phi'; T(\mathcal{R})) = -\alpha\, \Delta\big(T(\mathcal{R}), T(S_\phi)\big) + \beta\, \mathcal{U}_{\text{task}}(S_{\phi'})$, with $\alpha, \beta \ge 0$ and $\Delta, \mathcal{U}_{\text{task}}$ measurable, so that the expectation over $\phi' \sim \pi$ non-trivially integrates the task utility over posterior uncertainty.

- ***(Unbiasedness)*** $\mathbb{E}\big[g_i \mid \mathcal{F}_{i-1}\big] \in \partial_\phi^+ \mathcal{V}(\phi_{i-1})$*, where* ∂_ϕ^+ *denotes the superdifferential with respect to* ϕ*;*
- ***(Bounded variance)*** $\mathbb{E}\big[\|g_i\|^2 \mid \mathcal{F}_{i-1}\big] \le C$ *for some constant* $C < \infty$*.*

Let the update be $\phi_i = \Pi_\Phi\{\phi_{i-1} + \eta_i\, g_i\}$*, where* Π_Φ *is the projection onto a compact* convex *set* Φ*, with non-increasing step-sizes* $\eta_i > 0$ *satisfying* $\sum_i \eta_i = \infty$ *and* $\sum_i \eta_i^2 < \infty$*. Then* ϕ_i *converges almost surely to the set of stationary points of* $\mathcal{V}$ *on* Φ*, i.e.,*

$$\operatorname{dist}(\phi_i,\, \Phi_{\text{stat}}) \;\to\; 0 \quad a.s.,$$

where $\Phi_{\text{stat}} := \{\phi \in \Phi : 0 \in \partial_\phi^+ \mathcal{V}(\phi) + \mathcal{N}_\Phi(\phi)\}$ *and* $\mathcal{N}_\Phi(\phi)$ *is the normal cone of* Φ *at* ϕ*. If, moreover,* $\mathcal{V}$ *is strictly concave on* Φ*, then* $\Phi^\star = \{\phi^\star\}$ *is the unique stationary point and*

$$\operatorname{dist}(\phi_i, \Phi^\star) \;=\; \|\phi_i - \phi^\star\| \;\to\; 0 \quad a.s.$$

Proof 2 (Sketch) *Define the posterior expected utility* $\mathcal{V}(\phi) := \mathbb{E}_{\phi' \sim \pi(\cdot \mid T(\mathcal{R}))}[U(\phi, \phi';\, T(\mathcal{R}))]$*. Under assumptions (Unbiasedness) and (Bounded variance) together with the Robbins–Monro step-size conditions (Robbins and Monro, 1951), the projected stochastic supergradient ascent iterates* $\phi_i = \Pi_\Phi\{\phi_{i-1} + \eta_i g_i\}$ *satisfy the conditions of the projected stochastic approximation theorem. This guarantees almost sure convergence to the set of stationary points* Φ_{stat} *of* $\mathcal{V}$ *on the compact convex set* Φ*. Under strict concavity of* $\mathcal{V}$*, the set* Φ_{stat} *reduces to the unique global maximiser* $\phi^\star$*, so* $\phi_i \to \phi^\star$ *almost surely. In our setting,* g_i *is constructed from the feedback metrics* $\mathcal{F}_i$ *(statistical consistency, correlation preservation, and KS-test distances) using differentiable surrogates for* Δ *and* $\mathcal{U}_{\text{task}}$ *(e.g., smooth divergence approximations and differentiable proxy metrics). The (Unbiasedness) and (Bounded variance) assumptions hold provided these surrogates are unbiased estimators of* $\partial_\phi^+ \mathcal{V}$ *with uniformly bounded second moments — conditions that must be verified for any specific choice of surrogate.* ■

4 Experiments

4.1 Experimental Settings

Datasets. We use four real-world classification datasets from diverse domains, which are *Travel, Sick, Heloc,* and *Thyroid*, and four synthetic datasets with explicit inter-attribute correlations, which are *Consumer Behavior, Health Metrics, Real Estate*, and *Social Network*. Details of the datasets are provided in Appendix A.4. As with EPIC (Kim et al., 2024), each dataset is randomly split into 80% training and 20% test sets.

Comparative Methods. We compare RDDG against representative generative methods for tabular data synthesis, including GReaT (Borisov et al., 2022), EPIC (Kim et al., 2024), TabDDPM (Kotelnikov et al., 2022), CDTD (Mueller et al., 2025), REalTabFormer (Solatorio and Dupriez, 2023), and ADS-GAN (Yoon et al., 2020). We also report the vanilla classification performance without using any synthetic data, denoted as "Original".

Foundation Models. In our experiments, GPT-3.5 (GPT-3.5-turbo-0125) is used as our default LLM. Additionally, on real datasets, we will report the performance of RDDG alongside other representative LLMs, i.e., Llama 3.0 and Mistral Max.

Configurations. For core set selection, details about MLP implementation and training configurations are provided in Table 11 in Appendix A.4.

4.2 Experimental Results

4.2.1 Imbalanced Classification Results

We evaluate imbalanced classification using weighted Macro-F1, Balanced Accuracy (BAL ACC), Sensitivity, and Specificity metrics. In Appendix A.4, we provide details about these evaluation metrics. As with EPIC, we use four mainstream classification models —XGBoost (Chen and Guestrin, 2016), CatBoost (Dorogush et al., 2018), LightGBM (Ke et al., 2017), and GBDT — as baselines. The classifiers' performance is averaged to ensure robustness.

Results on the Real-world Datasets. Table 1 reports the classification results for four real-world datasets under imbalanced conditions. We observe that our proposed RDDG algorithm achieves strong BAL ACC and Sensitivity scores across most datasets. Moreover, in terms of the weighted Macro-F1 metric, RDDG also obtains the best performance on the Travel and Thyroid datasets. Heloc, unlike the other datasets, maintains class balance with approximately equal sample sizes. For such balanced datasets, BAL ACC is the most suitable evaluation metric, and RDDG outperforms all other methods on this metric. The only exception is the Sick dataset, where CDTD and TabDDPM obtain the best weighted Macro-F1 scores, yet RDDG still achieves the best Sensitivity score, which is

Dataset	Method	Macro-F1	BAL ACC	Sensitivity	Specificity
Travel	Original	58.12±2.04	71.21±1.56	56.48±2.81	85.63±0.85
	ADS-GAN	56.07±8.25	68.94±5.12	53.16±3.46	**88.25±3.31**
	REalTabFormer	53.25±4.10	67.70±2.69	58.42±4.19	86.82±1.31
	GReaT	60.95±2.59	72.86±1.80	58.80±3.69	86.92±0.79
	TabDDPM	65.32±1.95	73.19±2.15	71.98±2.89	84.14±1.56
	CDTD	66.32±2.13	74.82±0.91	72.06±2.19	85.23±3.12
	EPIC	66.65±2.53	78.23±2.10	78.00±4.59	78.46±2.50
	RDDG (Ours)	**68.63±2.12**	**79.67±4.68**	**78.23±1.23**	82.67±2.56
Sick	Original	87.82±2.46	91.22±0.93	82.84±1.76	99.61±0.28
	ADS-GAN	87.52±1.64	89.82±0.53	79.86±0.97	**99.82±0.16**
	REalTabFormer	85.17±2.04	89.30±1.16	79.02±2.28	99.57±0.12
	GReaT	87.23±1.86	90.83±1.09	82.06±2.10	99.60±0.09
	TabDDPM	88.16±2.79	91.89±1.52	84.24±2.90	99.55±0.17
	CDTD	**89.63±1.71**	93.25±0.97	86.96±1.87	99.53±0.12
	EPIC	85.08±1.89	92.45±0.68	85.98±1.31	98.93±0.27
	RDDG (Ours)	87.99±0.91	**93.62±0.95**	**88.04±1.93**	99.20±0.07
Heloc	Original	**71.01±0.47**	72.21±0.37	67.19±0.86	78.52±0.38
	ADS-GAN	70.00±0.29	71.08±0.24	67.68±0.28	78.52±0.62
	REalTabFormer	70.09±0.19	72.28±0.86	67.54±0.35	78.82±0.46
	GReaT	70.25±0.33	72.16±0.24	66.22±0.49	**79.70±0.21**
	TabDDPM	70.33±0.22	72.49±0.19	**67.89±0.39**	77.09±0.40
	CDTD	70.18±0.48	72.40±0.35	67.59±0.79	77.21±0.32
	EPIC	70.08±0.51	72.52±0.38	66.90±0.75	78.13±0.17
	RDDG (Ours)	70.32±0.55	**72.54±0.43**	67.72±0.78	77.35±0.13
Thyroid	Original	94.23±1.99	95.08±1.60	91.14±3.12	99.02±1.01
	ADS-GAN	76.40±6.58	81.14±4.55	62.27±9.10	**100.00±0.00**
	REalTabFormer	94.39±1.09	96.26±0.50	93.45±0.09	97.06±1.01
	GReaT	91.31±1.61	92.46±0.99	85.91±1.40	99.08±0.75
	TabDDPM	96.05±2.22	**96.94±0.77**	**94.74±0.00**	99.14±1.53
	CDTD	94.66±2.49	96.37±0.94	94.47±1.18	98.28±1.77
	EPIC	94.67±1.96	96.06±1.35	93.42±2.34	98.71±0.77
	RDDG (Ours)	**96.58±1.27**	96.71±1.17	93.42±2.34	99.95±0.00

Table 1: Imbalanced classification performance of different methods on four real datasets.

Dataset	Method	Macro-F1	BAL ACC	Sensitivity	Specificity
Consumer Behavior	Original	66.71±2.07	77.00±2.14	63.37±2.13	86.51±0.05
	ADS-GAN	65.77±1.12	77.53±1.08	61.27±1.04	84.43±1.03
	REalTabFormer	65.44±2.51	75.20±1.47	62.21±3.68	86.33±2.91
	GReaT	64.76±3.65	77.80±2.33	57.07±1.30	80.82±3.76
	TabDDPM	67.90±1.54	75.67±1.44	63.21±2.29	88.90±3.40
	CDTD	67.89±5.09	73.60±3.82	62.19±5.88	88.79±5.97
	EPIC	67.32±2.24	77.20±4.24	63.88±2.10	87.20±3.44
	RDDG (Ours)	**68.99±2.18**	**79.60±2.41**	**66.75±2.76**	**90.50±2.34**
Health Metrics	Original	90.95±1.35	95.38±1.05	84.42±1.43	93.07±1.27
	ADS-GAN	89.26±1.64	95.15±2.79	88.83±3.82	94.15±3.45
	REalTabFormer	93.42±2.34	95.33±2.66	92.64±1.42	96.25±2.27
	GReaT	93.29±2.49	96.12±2.57	92.72±1.36	96.40±1.78
	TabDDPM	**96.09±0.99**	96.00±1.22	**96.08±0.98**	**98.14±0.69**
	CDTD	95.49±0.78	95.50±0.95	95.47±0.79	97.77±0.53
	EPIC	93.89±2.41	95.10±2.89	91.56±1.90	95.56±2.21
	RDDG (Ours)	95.40±1.73	**96.74±2.21**	94.55±2.53	97.71±1.64
Real Estate	Original	80.06±1.32	83.97±1.28	79.75±1.44	92.75±1.17
	ADS-GAN	82.44±1.26	86.17±1.12	81.51±1.38	**95.51±1.35**
	REalTabFormer	81.18±2.26	84.97±2.45	80.74±2.54	90.74±2.19
	GReaT	85.16±1.26	88.38±1.28	82.03±3.27	92.03±2.83
	TabDDPM	76.00±3.06	87.87±3.17	**86.67±6.88**	89.08±0.71
	CDTD	72.98±3.41	85.34±3.06	81.90±6.29	88.77±0.54
	EPIC	85.21±2.10	86.98±4.17	83.45±4.32	93.21±3.42
	RDDG (Ours)	**88.70±1.72**	**88.50±1.12**	85.43±1.47	95.21±2.10
Social Network	Original	87.87±2.15	96.15±2.34	83.63±2.67	98.18±3.01
	ADS-GAN	88.89±1.28	96.20±1.06	83.63±2.19	97.18±2.68
	REalTabFormer	95.69±3.09	98.00±3.17	95.52±2.06	**99.49±3.58**
	GReaT	96.88±3.36	98.80±2.01	96.87±4.30	98.42±3.22
	TabDDPM	92.16±5.87	76.26±16.95	93.30±4.35	74.80±16.37
	CDTD	**98.99±0.89**	97.40±1.45	**99.00±0.89**	97.16±1.71
	EPIC	86.87±2.21	96.30±2.57	83.76±3.13	98.19±3.29
	RDDG (Ours)	97.12±2.13	**98.89±2.73**	97.66±2.17	99.24±3.21

Table 2: Imbalanced classification performance of different methods on four synthetic datasets.

an important criterion in imbalanced learning that assesses the performance on the minority classes.

Comparing RDDG with EPIC, we observe that on the imbalanced datasets, the average performance improvements are 2.27% in weighted Macro-F1, 1.09% in BAL ACC, 0.86% in Sensitivity, and 1.91% in Specificity, all of which are statistically significant.

Results on the Synthetic Datasets. Table 2 reports the classification performance on four synthetic datasets with explicit inter-attribute correlations. In Table 2, we observe that RDDG consistently obtains the best BAL ACC scores across all four datasets. It also achieves the best weighted Macro-F1 scores on the Consumer Behavior and Real Estate datasets, and ranks among the Top-2 on the Social Network dataset. The only exception is the Health Metrics dataset, in which TabDDPM and CDTD are the Top-2 performers, yet the gap between RDDG and CDTD is small. In terms of Sensitivity, RDDG is among the Top-2 performers on three out of the four datasets. On the Social Network dataset, we also observe a large variance in classification performance for TabDDPM, with performance varying substantially across classifiers, indicating that the utility of the generated samples differs across classification models.

Comparing RDDG with EPIC, the average performance improvements are 2.04% in weighted Macro-F1, 4.23% in BAL ACC, 5.44% in Sensitivity, and 2.13% in Specificity.

In Tables 3 and 4 in Appendix A.1, we also report the performance of EPIC and RDDG using other LLMs such as Llama 3.0 and Mistral Max, and the observations are generally consistent with GPT-3.5. Overall, these experiments validate the effectiveness of RDDG in generating high-quality samples for imbalanced classification.

4.2.2 Statistical Fidelity Evaluation

To comprehensively evaluate the statistical fidelity of synthetic data generated by EPIC and RDDG, we examine both distribution consistency and inter-attribute correlation preservation, by employing Kullback-Leibler (KL) divergence for the former, and Frobenius norm, Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) for the latter, more details are given in Appendix A.2.1.

The overall performance summary (Figure 2) shows RDDG's consistent advantages across all evaluated metrics, outperforming EPIC on distribution consistency in 6 of 8 datasets and demonstrating a uniform superiority in correlation preservation. A comprehensive analysis, including detailed visualizations and dataset-specific results, is provided in Appendix A.2.

Specifically, RDDG achieves an 18.2% reduc-

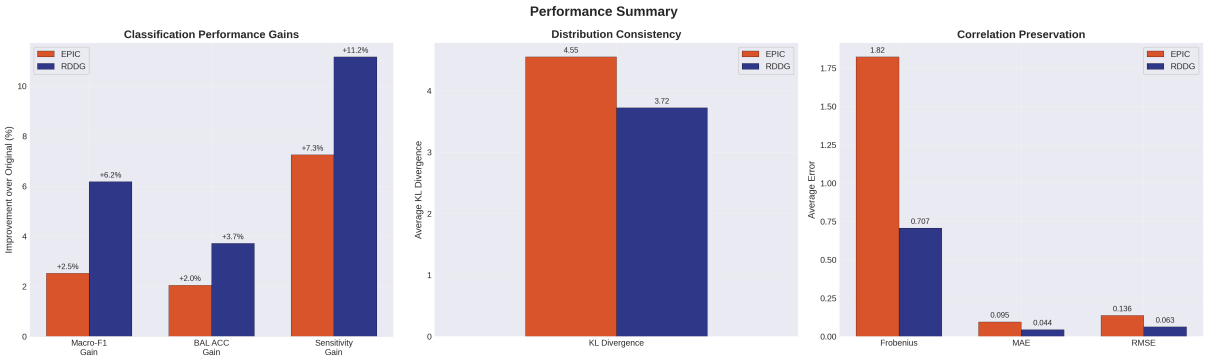


Figure 2: Overall performance summary comparing EPIC and RDDG across (a) classification performance gains (on the left, with higher values indicating better performance), (b) distribution consistency, and (c) correlation preservation metrics (center and right), where lower values indicate better performance.

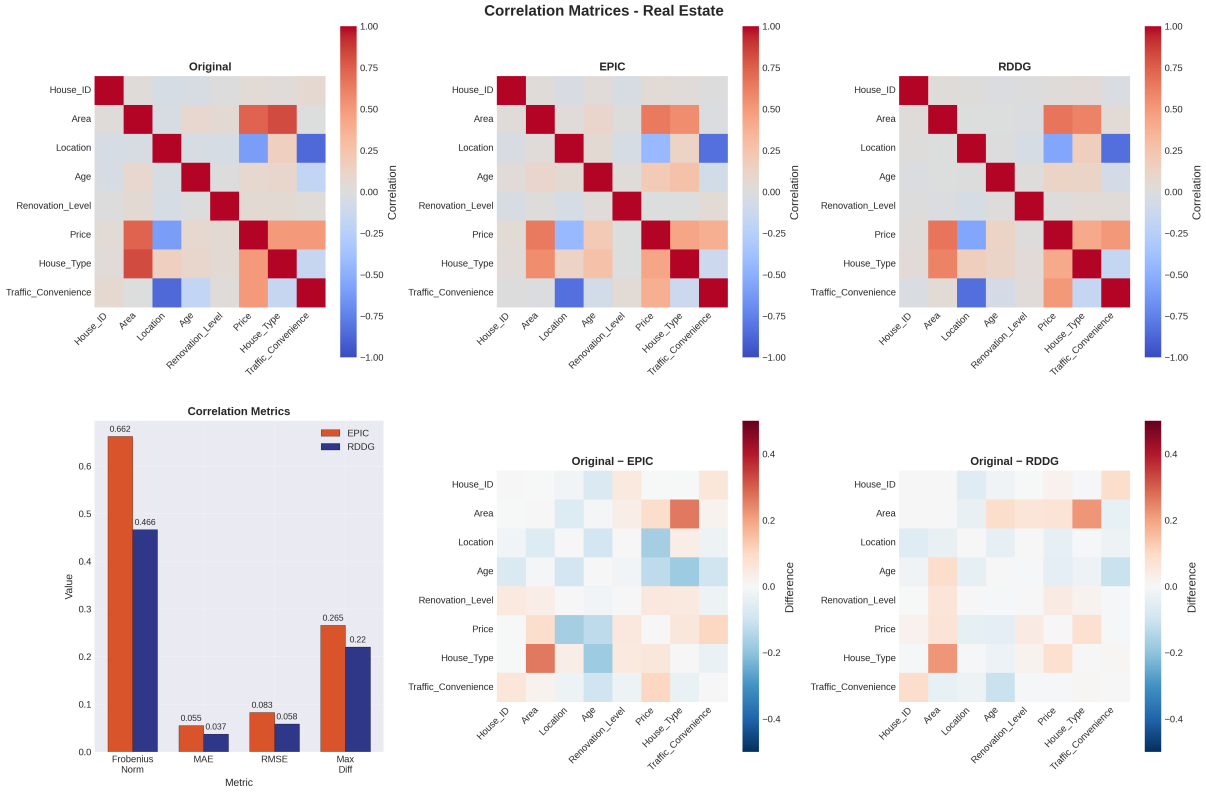


Figure 3: On the Real Estate dataset, RDDG demonstrates better correlation preservation than EPIC.

tion in overall KL divergence (3.72 vs. 4.55), indicating superior preservation of the original data's probabilistic structure (Figure 4 in A.2). The method demonstrates particularly strong performance in complex datasets, with notable improvements in the Thyroid dataset (90.2% reduction in KL divergence: 0.278 vs. 2.83) and Travel dataset (20.1% reduction: 16.3 vs. 20.4). Distribution comparisons across representative datasets illustrate RDDG's superior ability to capture complex distributional shapes and modalities (Figure 5).

In terms of correlation preservation, RDDG exhibits remarkable performance with a 65.6% improvement in overall Frobenius norm (0.827 vs. 2.40) and consistent advantages across all correlation metrics (Figures 6 and 7). The MAE analysis indicates that RDDG preserves correlation structures, achieving 58.1% higher accuracy than EPIC (0.048 vs. 0.115), and a 60.2% lower RMSE (0.065 vs. 0.163).

In particular, on datasets with explicit inter-attribute correlations, such as Real Estate, we show that RDDG better captures and preserves inter-attribute correlations than EPIC, as shown in Figure

3. Additional fidelity analysis results are provided in Appendix A.2.

4.3 Ablation Studies

Effects of Core Set and Feedback Mechanism To investigate the effect of the core set, we compare classification performance with randomly sampled subsets, on Travel and Thyroid. As shown in Table 6 in Appendix A.3, the core set algorithm achieves a substantial performance gain compared to randomly sampled subsets. Moreover, we also show the influence of our self-reinforcing feedback mechanism, which significantly improves the overall performance.

Influence of Imbalance Ratio To study the influence of imbalance ratio (IR) on the performance of EPIC and RDDG, we use the UCI segmentation dataset and artificially balance the ratios between the numbers of majority and minority class samples via an exponentially decaying strategy. As reported in Table 7 in Appendix A.3, RDDG consistently outperforms EPIC across all metrics for varying IR values. When the IR is high, the advantage of RDDG over EPIC is more substantial.

5 Conclusion

In this work, we propose a dynamically guided in-context tabular data synthesis framework that comprises progressive CoT steps and a self-reinforcing feedback mechanism. By integrating the explicit functional dependency constraints discovered through relation mining and the dynamic feedback mechanism, our framework significantly improves both data fidelity and downstream imbalanced classification performance.

Limitations

While RDDG demonstrates substantial improvements in tabular data synthesis, several limitations warrant consideration. First, our approach requires either external API calls or local deployment of a foundation model. Second, as an in-context learning framework, although it is training-free, our approach is bounded by the inherent capability of LLMs. Third, LLM token limitations constrain the volume of examples that can be processed simultaneously, necessitating batch-wise generation for large-scale synthesis tasks. However, this may affect the global consistency of generated samples across batches.

Acknowledgments

This work was partially supported by the MOE Liberal Arts and Social Sciences Foundation (No. 23YJAZH210), the Major Program of the National Social Science Foundation (No. 23&ZD309), the Henan Provincial Center for Outstanding Overseas Scientists (No. GZS2025004), and the High-Level Talent International Training Program of Henan Province (No. GCC2025010). Julian Rodemann acknowledges funding support from the Federal Statistical Office of Germany within the cooperation project "Machine Learning in Official Statistics", as well as from the Bavarian Institute for Digital Transformation (bidt) and the Bavarian Academy of Sciences and Humanities (BAdW) through a graduate scholarship. Esteban Garcés Arias acknowledges support from the Mentoring Program at the Faculty of Mathematics, Informatics, and Statistics at LMU Munich and from the MCML (Munich Center for Machine Learning). We thank the anonymous reviewers, area chairs, and program committee members for their constructive feedback.

Ethics Statement

We affirm that our research adheres to the ACL Ethics Policy. This work uses publicly available datasets and contains no personally identifiable information. We declare that there are no conflicts of interest that could potentially influence the outcomes, interpretations, or conclusions of this research. All funding sources supporting this study are acknowledged in the acknowledgments section. We have diligently documented our methodology, experiments, and results, and we commit to sharing our code, data, and other relevant resources to enhance reproducibility.

References

Sigrún Andradóttir and Vicki M Bier. 2000. Applying bayesian ideas in simulation. *Simulation Practice and Theory*, 8(3-4):253–280.

James O Berger. 2013. *Statistical decision theory and Bayesian analysis*. Springer Science & Business Media.

Vadim Borisov, Kathrin Seßler, Tobias Leemann, Martin Pawelczyk, and Gjergji Kasneci. 2022. Language models are realistic tabular data generators. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. 2002. SMOTE: Synthetic minority over-sampling technique. *Preprint*, arXiv:1106.1813.

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*, pages 785–794. ACM.

Anna Veronika Dorogush, Vasily Ershov, and Andrey Gulin. 2018. CatBoost: Gradient boosting with categorical features support. In *Proceedings of the 2018 IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 1021–1028. IEEE.

Josh Gardner, Juan C. Perdomo, and Ludwig Schmidt. 2024. Large scale transfer learning for tabular data via language modeling. In *Advances in Neural Information Processing Systems (NeurIPS 2024)*, pages 45155–45205.

Noah Hollmann, Samuel Müller, Katharina Eggensperger, and Frank Hutter. 2023. Tabpfn: A transformer that solves small tabular classification problems in a second. In *The Eleventh International Conference on Learning Representations (ICLR 2023)*.

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.

Yuxin Hong, Xiao Zhang, Xin Zhang, and Joey Tianyi Zhou. 2024. Evolution-aware variance (EVA) coreset selection for medical image classification. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 301–310.

Haque Ishfaq, Assaf Hoogi, and Daniel L. Rubin. 2018. TVAE: Triplet-based variational autoencoder using metric learning. *Preprint*, arXiv:1802.04403.

Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. 2017. LightGBM: A highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems*, pages 3146–3154. Curran Associates Inc.

Jinhee Kim, Taesung Kim, and Jaegul Choo. 2024. EPIC: Effective prompting for imbalanced-class data synthesis in tabular data classification via large language models. In *Advances in Neural Information Processing Systems*.

Akim Kotelnikov, Dmitry Baranchuk, Ivan Rubachev, and Artem Babenko. 2022. TabDDPM: Modelling tabular data with diffusion models. *Preprint*, arXiv:2209.15421.

Xun Liang, Hanyu Wang, Yezhaohui Wang, Shichao Song, Jiawei Yang, Simin Niu, Jie Hu, Dan Liu, Shunyu Yao, Feiyu Xiong, and 1 others. 2024. Controllable text generation for large language models: A survey. *arXiv preprint arXiv:2408.12599*.

Jialun Liu, Yifan Sun, Chuchu Han, Zhaopeng Dou, and Wenhui Li. 2020. Deep representation learning on long-tailed data: A learnable embedding augmentation perspective. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition (CVPR 2020)*, pages 2970–2979.

Markus Mueller, Kathrin Gruber, and Dennis Fok. 2025. Continuous diffusion for mixed-type tabular data. In *The Thirteenth International Conference on Learning Representations (ICLR 2025)*.

Noseong Park, Mahmoud Mohammadi, Kshitij Gorde, Sushil Jajodia, Hongkyu Park, and Youngmin Kim. 2018. Data synthesis based on generative adversarial networks. *Proceedings of the VLDB Endowment*, 11:1071–1083.

David Poole and Adrian E Raftery. 2000. Inference for deterministic simulation models: the bayesian melding approach. *Journal of the American Statistical Association*, 95(452):1244–1255.

Herbert Robbins and Sutton Monro. 1951. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407.

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

Boris van Breugel and Mihaela van der Schaar. 2024. Position: Why tabular foundation models should be a research priority. In *Forty-first International Conference on Machine Learning (ICML 2024)*.

Stephen Wade, Marianne F Weber, Peter Sarich, Pavla Vaneckova, Silvia Behar-Harpaz, Preston J Ngo, Sonya Cressman, Coral E Gartner, John M Murray, and Tony A Blakely. 2022. Bayesian calibration of simulation models: a tutorial and an australian smoking behaviour model. *arXiv preprint arXiv:2202.02923*.

Chaozheng Wang, Shuzheng Gao, Pengyun Wang, Cuiyun Gao, Wenjie Pei, Lujia Pan, and Zenglin Xu. 2022. Label-aware distribution calibration for long-tailed classification. *IEEE Transactions on Neural Networks and Learning Systems*, 35(5):6963–6975.

Wentao Wang, Suhang Wang, Wenqi Fan, Zitao Liu, and Jiliang Tang. 2020. Global-and-local aware data generation for the class imbalance problem. In *Proceedings of the 2020 SIAM International Conference on Data Mining (SDM 2020)*, pages 307–315.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V. Le, and Denny

Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837.

Xumeng Wen, Han Zhang, Shun Zheng, Wei Xu, and Jiang Bian. 2024. From supervised to generative: A novel paradigm for tabular deep learning with large language models. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD 2024)*, pages 3323–3333.

Lei Xu, Maria Skoularidou, Alfredo Cuesta-Infante, and Kalyan Veeramachaneni. 2019. Modeling tabular data using conditional GAN. In *Advances in Neural Information Processing Systems*.

Jinsung Yoon, Louise N. Drumright, and Mihaela van der Schaar. 2020. Anonymization through data synthesis using generative adversarial networks (ADS-GAN). *IEEE Journal of Biomedical and Health Informatics*, 24(8):2378–2388.

Hanqing Zhang, Haolin Song, Shaoyu Li, Ming Zhou, and Dawei Song. 2023a. A survey of controllable text generation using transformer-based pre-trained language models. *ACM Computing Surveys*, 56(3):1–37.

Hengrui Zhang, Jiani Zhang, Balasubramaniam Srinivasan, Zhengyuan Shen, Xiao Qin, Christos Faloutsos, Huzefa Rangwala, and George Karypis. 2023b. Mixed-type tabular data synthesis with score-based diffusion in latent space. *arXiv preprint arXiv:2310.09656*.

Qihao Zhao, Yalun Dai, Hao Li, Wei Hu, Fan Zhang, and Jun Liu. 2024a. LTGC: Long-tail recognition via leveraging LLMs-driven generated content. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19510–19520.

Zilong Zhao, Aditya Kunar, Robert Birke, Hiek van der Scheer, and Lydia Y. Chen. 2024b. CTAB-GAN+: Enhancing tabular data synthesis. *Frontiers in Big Data*, 6:1296508.

A Appendix

A.1 Additional Imbalanced Classification Results

In our experiments, GPT-3.5 (GPT-3.5-turbo-0125) is used as our default LLM. While the original EPIC study (Kim et al., 2024) used GPT-3.5-turbo-0613 and GPT-3.5-turbo-16k-0613 models, in this work, we evaluate EPIC and our approach using GPT-3.5-turbo-0125, as it is the model currently available to us[2]. In addition to GPT-3.5, we also report the performance of EPIC and RDDG using other LLMs, such as Llama 3.0 and Mistral Max, and present the results below. For each dataset, each method will generate *1000* synthetic samples (the target threshold).

Impact of LLM Choice on the Imbalanced Classification Performance. To investigate the impact of LLM choice on RDDG performance, we conduct additional classification experiments that incorporate the open-source LLMs Llama 3.0 and Mistral Max, as well as the proprietary GPT-3.5 model used in our approach. As shown in Table 3, we evaluate our RDDG framework using these alternative LLMs for comparison. A comparative analysis reveals that Mistral Max underperforms the other two LLMs in classification accuracy, with Llama 3.0 slightly trailing GPT-3.5. In Table 4, we also compare the performance of EPIC and RDDG under Llama 3.0 and Mistral Max. We observe that RDDG consistently outperforms EPIC across both LLMs, except for Thyroid when using Mistral. Overall, regardless of LLM choice, RDDG outperforms EPIC.

Dataset	Method	#syn	Macro-F1	BAL ACC	Sensitivity	Specificity
Travel	Original	-	58.12±2.04	71.21±1.56	58.12±2.04	**85.63 ± 0.85**
	Mistral	1K	66.21±1.32	78.01±1.72	77.21±1.02	76.21±1.99
	Llama 3.0	1K	65.32±1.02	77.23±2.35	76.97±0.71	77.15±1.56
	GPT-3.5	1K	**68.51±2.11**	**79.52±3.16**	**79.16±0.11**	83.55±2.64
Sick	Original	-	87.82±2.46	91.22±0.93	82.84±1.76	**99.61 ± 0.28**
	Mistral	1K	88.04±1.36	94.62±1.53	**90.23±3.35**	99.02±0.36
	Llama 3.0	1K	**88.76±1.67**	**94.67±0.50**	90.22±1.12	99.13±0.28
	GPT-3.5	1K	87.99±0.91	93.62±0.95	88.04 ±1.93	99.20±0.07
Heloc	Original	-	**71.01±0.47**	72.21±0.37	67.19±0.86	**78.52±0.38**
	Mistral	1K	70.46±0.55	72.53±0.38	68.25±0.91	76.81±0.19
	Llama 3.0	1K	70.62±0.56	**72.63±0.41**	**68.53±0.89**	76.74±0.29
	GPT-3.5	1K	70.32±0.55	72.54±0.43	67.72±0.78	77.35±0.13
Thyroid	Original	-	94.23±1.99	95.08±1.60	91.14±3.12	99.02±1.01
	Mistral	1K	93.23±3.15	94.75±2.29	90.79±4.48	98.71±1.47
	Llama 3.0	1K	94.06±2.22	95.85±1.33	93.42±2.34	98.28±1.25
	GPT-3.5	1K	**96.58±1.27**	**96.71±1.17**	**93.42±2.34**	**99.95±0.00**

Table 3: Ablation study on the impact of LLM choice on RDDG performance.

Dataset	Method	Llama				Mistral			
		Macro-F1	BAL ACC	Sensitivity	Specificity	Macro-F1	BAL ACC	Sensitivity	Specificity
Travel	EPIC	63.24±1.22	74.23±3.20	74.67±1.21	75.25±1.29	64.35±0.98	75.32±2.21	76.78±1.22	75.67±1.48
	RDDG	**65.32±1.02**	**77.23±2.35**	**76.97±0.71**	**77.15±1.56**	**66.21±1.32**	**78.01±1.72**	**77.21±1.02**	**76.21±1.99**
Sick	EPIC	84.41±1.36	93.44±0.84	88.26±1.64	98.62±0.13	86.43±1.05	94.50±0.49	90.22±1.12	98.77±0.22
	RDDG	**88.76±1.67**	**94.67±0.50**	**90.22±1.12**	**99.13±0.28**	**88.04±1.36**	**94.62±1.53**	**90.22±3.35**	**99.02±0.36**
Heloc	EPIC	70.18±0.30	72.29±0.23	67.91±0.57	76.66±0.49	70.14±0.16	72.40±0.09	67.47±0.69	**77.32±0.80**
	RDDG	**70.62±0.56**	**72.63±0.41**	**68.53±0.89**	**76.74±0.29**	**70.46±0.55**	**72.53±0.38**	**68.25±0.91**	76.81±0.19
Thyroid	EPIC	88.96±2.54	91.24±0.88	84.21±0.00	98.28±1.77	**95.38±1.14**	**96.72±0.38**	**94.74±0.00**	**98.71±0.77**
	RDDG	**94.06±2.22**	**95.85±1.33**	**93.42±2.34**	**98.28±1.25**	93.23±3.15	94.75±2.29	90.79±4.48	98.71±1.47

Table 4: Comparison of EPIC and RDDG under Llama 3.0 and Mistral Max.

LLM Cost. While LLM cost is a valid concern, our findings in Table 5 show that it remains manageable in practice. "Expenses" denotes the API cost for each dataset. For instance, generating 1,000 new samples costs under 0.5$ using the GPT API. "Token" represents the total number of input and output tokens for each approach. Note that output tokens typically cost three to five times more than input tokens, as

[2]https://platform.openai.com/docs/deprecations/2023-11-06-chat-model-updates. As of June 17, 2024, GPT-3.5-turbo-0613 and GPT-3.5-turbo-16k-0613 have been deprecated by OpenAI, and "only existing users of these models will be able to continue using them".

Dataset	Method	Macro-F1	BAL ACC	Sensitivity	Specificity	Token	Expenses	Prompt 1	Prompt 2	Prompt 3	All
Consumer Behavior	EPIC	67.32±2.24	77.20±4.24	63.88±2.10	87.20±3.44	186K	$0.13	-	-	-	76.0s
	RDDG	68.99±2.18	79.60±2.41	66.75±2.76	90.50±2.34	176K	$0.12	6.0s	2.2s	137.3s	169.7s
Health Metrics	EPIC	93.89±2.41	95.10±2.89	91.56±1.90	95.56±2.21	475K	$0.43	-	-	-	96.3s
	RDDG	95.40±1.73	96.74±2.21	94.55±2.53	97.71±1.64	493K	$0.34	6.4s	2.7s	47.6s	78.7s
Real Estate	EPIC	85.21±2.10	86.98±1.47	83.45±4.32	93.21±4.02	150K	$0.10	-	-	-	35.2s
	RDDG	88.70±1.72	88.50±1.12	85.43±1.47	95.21±2.10	170K	$0.10	5.7s	2.4s	37.8s	66.7s
Social Network	EPIC	86.87±2.21	96.30±2.57	83.76±3.13	98.19±3.29	330K	$0.15	-	-	-	32.7s
	RDDG	97.12±2.13	98.89±2.73	97.66±2.17	99.24±3.21	240K	$0.17	6.0s	2.7s	38.9s	58.5s

Table 5: Comparison of EPIC and RDDG in terms of accuracy performances, number of input and output tokens, API expenses, and running time.

generating responses is computationally more intensive than processing prompts. It is also worth noting that both EPIC and RDDG remove the requirement for computationally expensive model training. That is, they enable users to generate synthetic samples directly through the GPT API or locally deployed open-source LLMs, eliminating the need for model training or fine-tuning.

Time Efficiency. In Table 5, we also report the total and specific time consumption of different prompts in RDDG, in which we find that Prompt 3 consumes substantially more time since it involves batch-wise iterative data generation and a self-reinforcing feedback mechanism. Despite the increased time required, RDDG consistently outperforms EPIC, producing higher-quality synthetic data. Overall, because both methods eliminate the need for computationally expensive model training, they exhibit outstanding efficiency (under 170 seconds) compared to other non-in-context learning approaches, which typically require multiple hours or even days of model training or fine-tuning.

A.2 Fidelity Analysis

To evaluate the fidelity of the synthetic data generated by EPIC and RDDG methods, we conduct an analysis focusing on two critical aspects: (i) distribution consistency of the synthetic data with respect to the original data, and (ii) preservation of inter-attribute correlations. The evaluations are performed across all datasets using standardized preprocessing and normalization procedures.

A.2.1 Evaluation Metrics for Statistical Fidelity

We employ several metrics to assess the statistical fidelity of synthetic data along two critical dimensions: distributional consistency and preservation of inter-attribute correlations. For distribution consistency, we employ the Kullback-Leibler (KL) divergence as our primary metric, which quantifies the information loss incurred when approximating the original distribution. For inter-attribute correlation preservation, we first calculate the inter-attribute Pearson correlation coefficients in the original data and the synthetic data generated by EPIC and RDDG, respectively, and then derive the correlation difference matrix (taking absolute value) between the correlation coefficient matrices of the original data and the synthetic data generated by EPIC and RDDG, respectively. We then use four metrics: Frobenius norm, Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and Max Difference to measure the overall preservation of inter-attribute correlations in the correlation difference matrix. Below, we provide formal definitions of each metric.

Kullback-Leibler (KL) Divergence The KL divergence quantifies the information loss when approximating the original data distribution P with the synthetic data distribution Q. For continuous attributes, we discretize the data into k bins and compute:

$$D_{KL}(P\|Q) = \sum_{i=1}^{k} P(i) \log \frac{P(i)}{Q(i)} \tag{6}$$

where $P(i)$ and $Q(i)$ represent the probability mass in bin i for the original and synthetic distributions, respectively. In our experiments, we use $k = 50$ bins with equal-width binning after standardization. The overall KL divergence for a dataset is computed as the mean across all numeric attributes:

$$\overline{D}_{KL} = \frac{1}{n} \sum_{j=1}^{n} D_{KL}(P_j\|Q_j) \tag{7}$$

where n is the number of numeric attributes. Lower values indicate better preservation of the distribution, with $D_{KL} = 0$ indicating a perfect match.

Frobenius Norm Let $\mathbf{C}_{real} \in \mathbb{R}^{n \times n}$ and $\mathbf{C}_{syn} \in \mathbb{R}^{n \times n}$ denote the Pearson correlation matrices for the original and synthetic data, respectively, where each element C_{ij} represents the correlation coefficient between attributes i and j.

The Frobenius norm measures the overall magnitude of differences between correlation matrices (i.e., the correlation difference matrix):

$$\|\mathbf{C}_{real} - \mathbf{C}_{syn}\|_F = \sqrt{\sum_{i=1}^{n}\sum_{j=1}^{n} |C_{real,ij} - C_{syn,ij}|^2} \tag{8}$$

This metric provides a single scalar value that quantifies the total deviation in the correlation structure.

Mean Absolute Error (MAE) The MAE computes the average absolute difference across all pairwise correlations:

$$\text{MAE} = \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n} |C_{real,ij} - C_{syn,ij}| \tag{9}$$

This metric is more interpretable than the Frobenius norm, as it represents the average deviation of the correlation coefficients.

Root Mean Square Error (RMSE) The RMSE penalizes larger deviations more heavily than MAE:

$$\text{RMSE} = \sqrt{\frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n} (C_{real,ij} - C_{syn,ij})^2} \tag{10}$$

Maximum Absolute Difference (Max Diff) The maximum absolute difference identifies the worst-case correlation preservation:

$$\text{Max Diff} = \max_{i,j} |C_{real,ij} - C_{syn,ij}| \tag{11}$$

This metric helps identify worst-case scenarios for specific attribute pairs where correlation preservation fails substantially.

For all metrics described above, lower values indicate better preservation of statistical properties. Specifically, in our evaluations:

- **KL Divergence**: Values near 0 indicate excellent distribution matching; values above 1.0 suggest substantial distributional differences.
- **Correlation Metrics**: For datasets with n attributes, perfect correlation preservation yields 0 for all metrics. As a reference, random synthetic data typically produces Frobenius norm values of $O(\sqrt{n})$.
- **Comparative Analysis**: We report both absolute metrics and relative improvements (percentage reduction) compared to baseline methods to contextualize performance gains.

A.2.2 Distribution Consistency

Within each dataset, we compute the mean KL divergence over numeric attributes and compute macro-averages across datasets using equal weighting. Figure 4 presents the comparative KL divergence analysis across all eight datasets. We observe that RDDG outperforms EPIC on 6 of 8 datasets. Notable improvements are observed on the Travel dataset (16.3 vs. 20.4, a 20.1% reduction), the Thyroid dataset (KL divergence: 0.278 for RDDG vs. 2.83 for EPIC, a 90.2% reduction), and the Consumer Behavior dataset (2.37 vs. 3.5, a 32.3% reduction). The Heloc and Social Network datasets are exceptions where EPIC demonstrates better performance. Overall, RDDG demonstrates superior distributional consistency

with the original data, with a macro-average KL divergence of 3.72 compared to EPIC's 4.55, representing an 18.2% improvement.

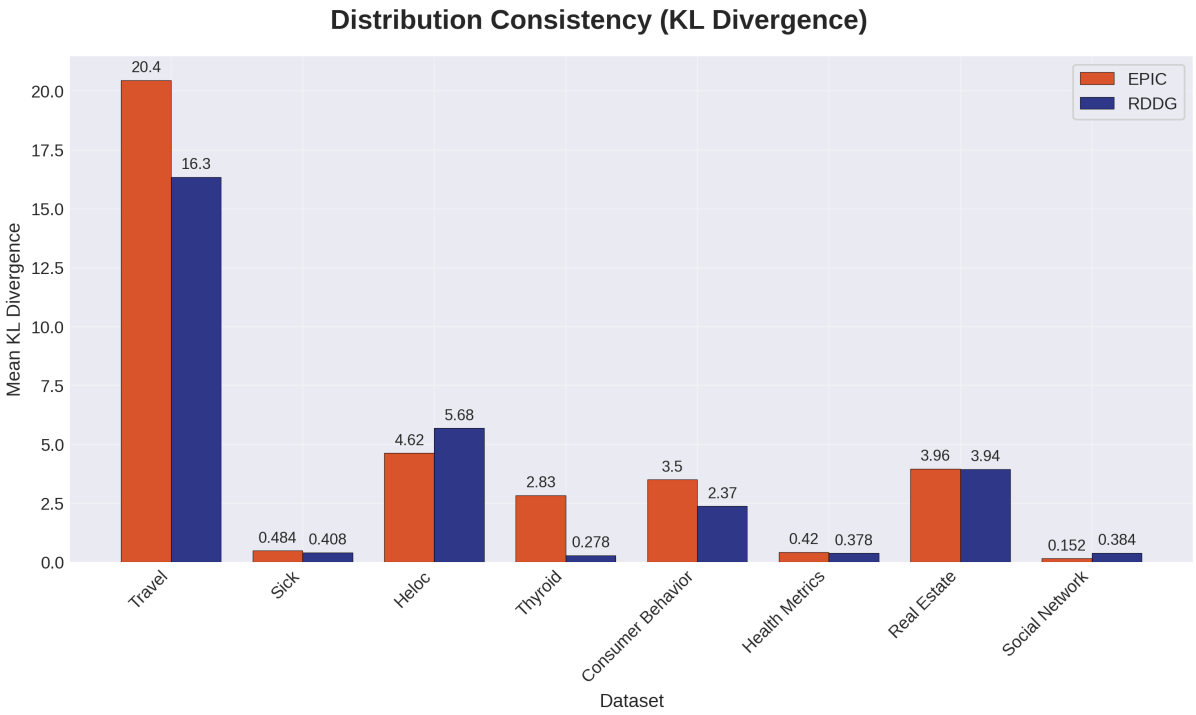
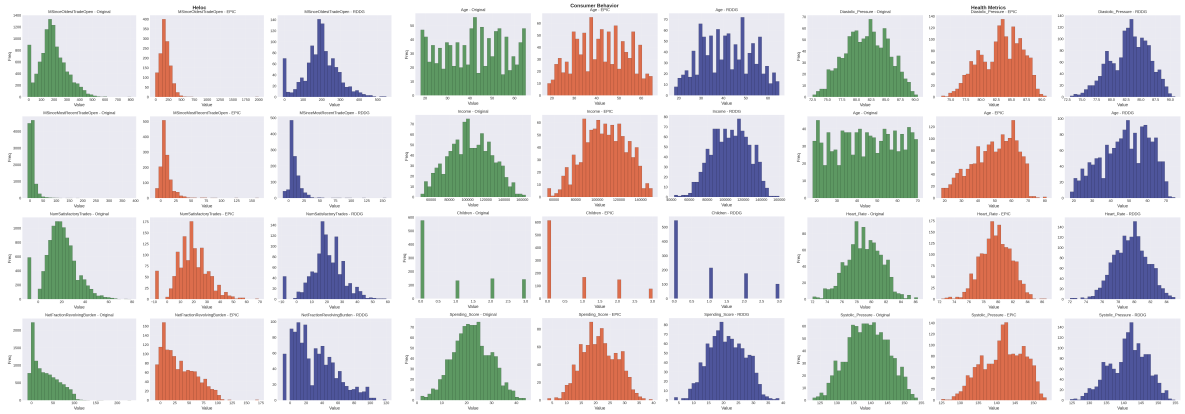


Figure 4: Mean KL divergence per dataset comparing EPIC and RDDG methods. Lower values indicate better distribution preservation.



(a) Heloc (b) Consumer Behavior (c) Health Metrics

Figure 5: Distribution comparisons between original data, and synthetic data generated by both EPIC and RDDG, respectively, over selected features/attributes across three representative datasets.

To provide detailed insights into distribution preservation, we visualize the distributions on representative datasets in Figures 5a– 5c. We observe that the synthetic data generated by both EPIC and RDDG generally follows the distribution of the attributes in the original datasets.

A.2.3 Inter-Attribute Correlation Preservation

Preserving correlation structures is crucial for maintaining data fidelity in synthetic data. Figure 6 presents the correlation preservation analysis results on Thyroid. The first three subfigures display the inter-attribute Pearson correlation coefficients for the original data and the two synthetic datasets from EPIC and RDDG, respectively. The last two subfigures show the correlation difference matrix between the original

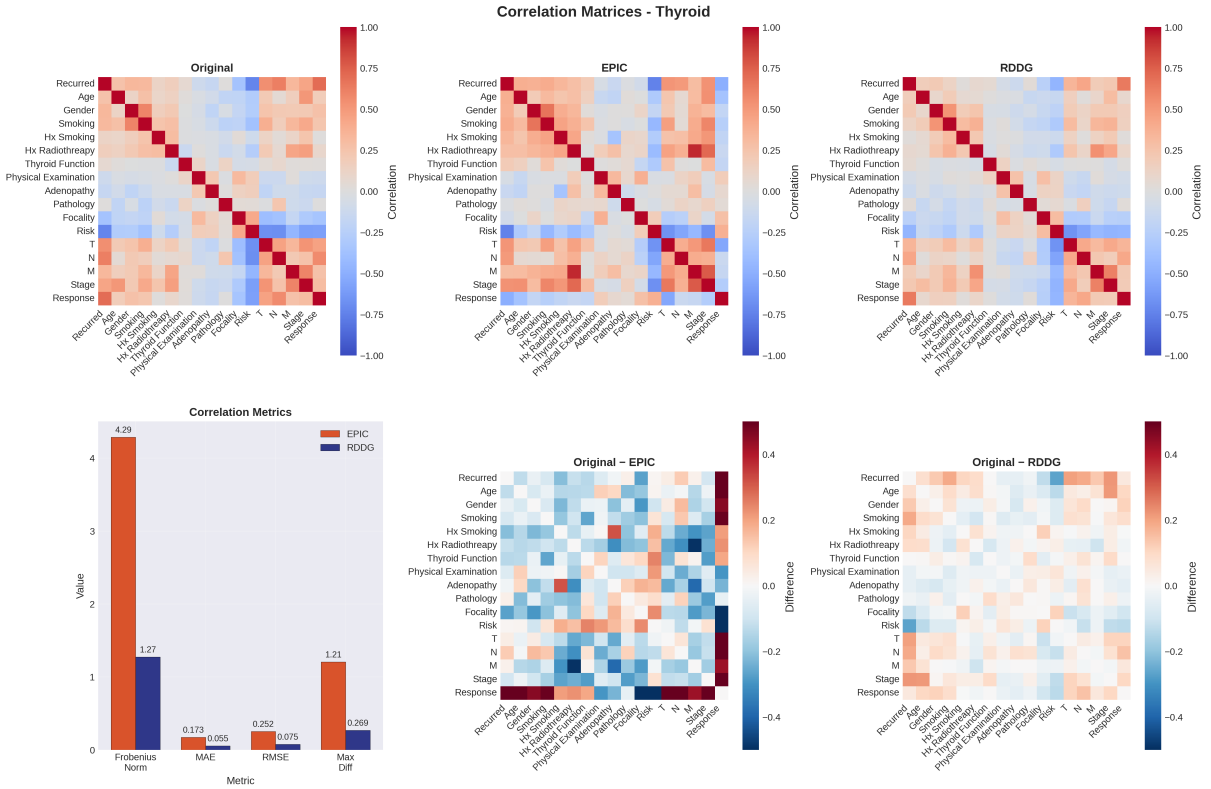


Figure 6: Correlation matrix analysis for the Thyroid dataset showing original correlations, synthetic data correlations (EPIC and RDDG), difference heatmaps, and preservation metrics comparison.

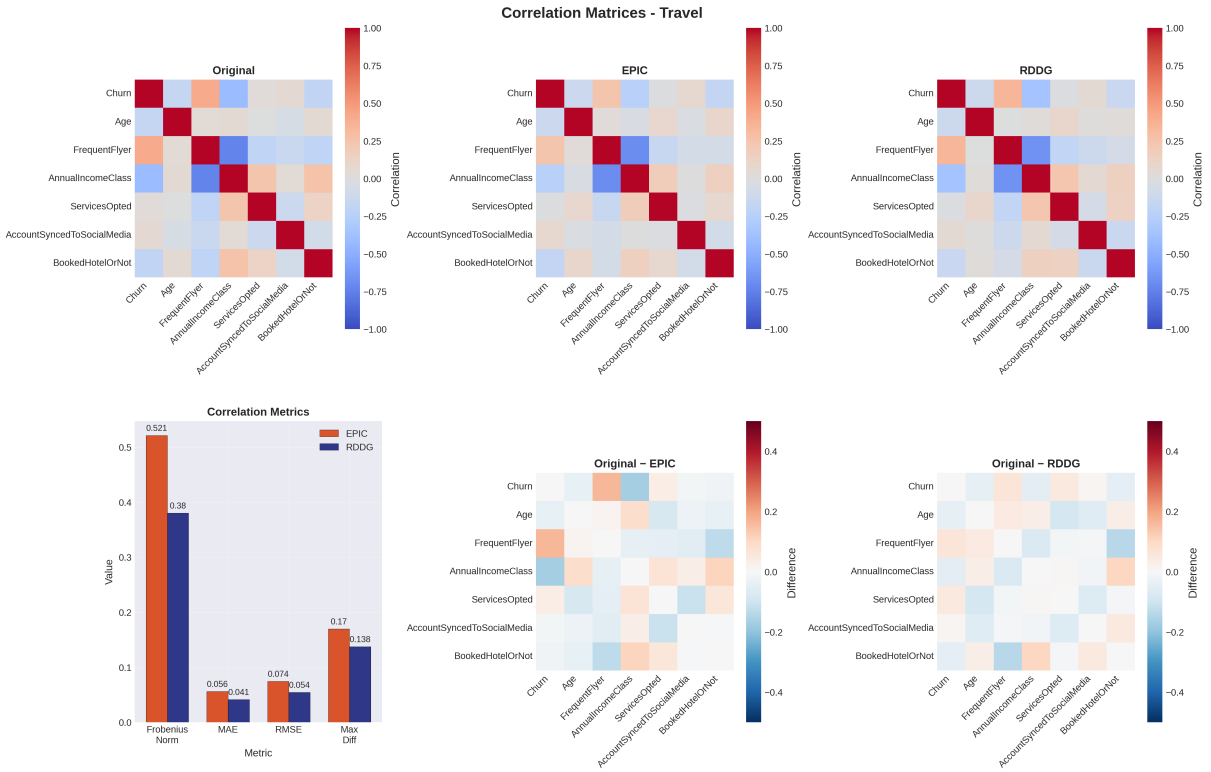


Figure 7: Correlation matrix analysis for the Travel dataset demonstrating superior correlation preservation by RDDG across all evaluation metrics.

data and each of the two synthetic datasets, with lighter shades indicating better preservation of inter-attribute correlations. We observe that RDDG exhibits substantial superiority in preserving inter-attribute correlations. Finally, the fourth subfigure calculates the overall inter-attribute correlation preservation degree using Frobenius norm, MAE, RMSE, and Max Diff, showing remarkable improvements over RDDG: Frobenius norm of 1.27 versus EPIC's 4.29 (70.4% improvement), MAE of 0.065 versus 0.173 (62.4% improvement), RMSE of 0.075 versus 0.252 (70.2% improvement), and Max Diff of 0.269 versus 1.21 (77.8% improvement).

Similar observations also hold for the Travel and Real Estate datasets (Figures 7 and 3), where RDDG consistently outperforms EPIC across all correlation preservation metrics, maintaining inter-attribute structural relationships with substantially higher fidelity. These results indicate that RDDG preserves both distributional properties and inter-attribute relationships more effectively than EPIC.

A.3 Detailed Ablation Study Results

To investigate the impact of core set selection, we compare classification performance between core sets and randomly sampled subsets on the Travel and Thyroid datasets. Note that the reference set size is set to 20 for the Thyroid dataset in this set of ablation studies. As shown in Table 6, the core set algorithm achieves substantial performance gains over random sampling. Furthermore, we demonstrate the effectiveness of our dynamic guidance (self-reinforcing feedback) mechanism, which significantly enhances overall performance, as evidenced in Table 6.

To examine how the imbalance ratio (IR) affects the performance of EPIC and RDDG, we use the UCI Segmentation dataset[3] and artificially vary the ratio between majority- and minority-class samples using an exponentially decaying strategy. As reported in Table 7, RDDG consistently outperforms EPIC across all metrics under varying IR levels, and its advantage over EPIC becomes more obvious as IR increases.

Dataset	Method	Macro-F1	BAL ACC	Sensitivity	Specificity
Travel	RDDG	68.51±2.11	79.52±5.16	79.16±0.11	83.55±2.64
	RDDG w/o Feedback Mechanism	66.91±2.10	77.21±1.21	78.12±2.35	82.23±1.34
	RDDG w/o CoreSet	67.55±1.32	78.13±2.47	79.00±1.32	82.96±2.76
Thyroid	RDDG	97.30±1.31	97.37±0.44	94.74±0.00	100.00±0.88
	RDDG w/o Feedback Mechanism	94.74±0.00	96.51±0.00	94.74±0.00	98.28±0.00
	RDDG w/o CoreSet	96.02±1.31	96.94±0.44	94.74±0.00	99.14±0.88

Table 6: Ablation study on Travel and Thyroid datasets.

Dataset	IR	Method	Macro-F1	BAL ACC	Sensitivity	Specificity	Token	Expenses	Prompt 1	Prompt 2	Prompt 3	All
Segmentation	2	EPIC	95.65±1.55	95.64±1.56	95.64±1.56	98.61±0.26	360K	$0.27	-	-	-	312.4s
		RDDG	96.32±1.95	96.29±1.96	96.29±1.96	99.38±0.33	317K	$0.25	7.4s	2.3s	231.5s	279.6s
	5	EPIC	95.80±1.41	95.76±1.43	95.76±1.43	99.29±0.24	400K	$0.32	-	-	-	321.2s
		RDDG	96.01±1.29	95.98±1.31	95.98±1.31	99.33±0.22	320K	$0.26	4.8s	4.8s	263.9s	295.8s
	10	EPIC	94.43±1.29	94.40±1.29	94.40±1.29	99.07±0.22	330K	$0.26	-	-	-	269.3s
		RDDG	95.62±0.90	95.55±0.93	95.55±0.93	99.26±0.16	360K	$0.29	4.9s	2.7s	341.0s	358.4s
	20	EPIC	90.53±1.93	91.57±1.88	91.57±1.88	97.93±0.31	370K	$0.20	-	-	-	284.7s
		RDDG	91.98±1.20	92.02±1.20	92.02±1.20	98.67±0.20	1343K	$0.81	4.9s	2.1s	719.9s	746.8s

Table 7: Ablation study results on the Segmentation dataset under different imbalance ratios (IR).

Dataset	Ref. size	Macro-F1	BAL ACC	Sensitivity	Specificity	Token	Expenses	Prompt 1	Prompt 2	Prompt 3	All
Travel	5	64.23±2.97	75.23±5.89	72.23±1.98	**84.65±2.34**	110K	$0.09	5.1s	2.2s	43.0s	61.0s
	10	68.51±2.11	79.52±5.16	**79.16±0.11**	83.55±2.64	160K	$0.10	5.7s	2.3s	30.4s	51.2s
	20	66.23±1.23	79.23±5.23	78.99±1.21	83.25±1.43	130K	$0.11	6.9s	3.1s	14.2s	34.1s
	30	**68.63±2.12**	**79.67±4.68**	78.23±1.23	82.67±2.56	190K	$0.13	6.1s	2.1s	23.4s	41.7s
Thyroid	5	95.38±1.14	96.72±0.38	94.72±0.00	98.71±0.77	358K	$0.24	5.9s	2.5s	150.5s	172.7s
	10	95.41±2.13	96.73±0.73	94.73±0.00	98.71±1.47	353K	$0.24	5.9s	2.3s	107.2s	127.3s
	20	**97.30±1.31**	**97.30±0.44**	**94.74±0.00**	**100.00±0.88**	352K	$0.24	8.4s	2.3s	99.9s	126.2s
	30	96.58±1.27	96.71±1.17	93.42±2.34	99.95±0.00	440K	$0.29	6.2s	2.6s	82.1s	100.9s

Table 8: Ablation study on the influence of reference set size on the overall performance.

In Table 8, we also study the influence of the reference set size (*Ref. size*) on the overall performance of RDDG. As with EPIC, the default size is fixed to 30 (15 samples per class) for all four real datasets. We

[3]https://archive.ics.uci.edu/dataset/50/image+segmentation.

show that it can be manually tuned for different datasets to further improve the overall performance.

A.4 Datasets, Evaluation Metrics and Implementation Details

Datasets. Tables 9 and 10 present the statistics and descriptions of the datasets and their attributes. For the four synthetic datasets, to study the preservation of inter-attribute correlations, we manually define correlations among some attributes before generating the corresponding values. For instance, in the *Real Estate* dataset, the *price* of an apartment is defined to be the multiplication of *basic price per square meter*, *apartment size*, and *renovation level*, minus age discount. In our code repository, we also provide code to generate these four datasets.

Dataset	Num. attributes	Num. samples	Num. classes	Class samples	IR
Travel	6	894	2	{0: 678, 1: 216}	3.139
Sick	27	3711	2	{'negative': 3480, 'sick': 231}	15.065
Heloc	23	10459	2	{'Bad': 4364, 'Good': 4003}	1.090
Thyroid	16	383	2	{'No': 275, 'Yes': 108}	2.546
Consumer Behavior	9	1000	2	{Home: 518, Food: 482}	1.075
Health Metrics	9	1000	3	{low risk: 500, medium risk: 300, high risk: 200}	2.500
Real Estate	8	1000	2	{no: 788, yes: 212}	3.717
Social Network	9	1000	4	{0: 789, 1: 86, 3: 70, 2: 55}	14.345

Table 9: Overview of representative dataset statistics.

Dataset	Attributes
Sick	The Sick dataset contains patient features such as age, sex, and thyroid-related test indicators, along with corresponding diagnosis results (sick or negative).
Travel	The Travel dataset contains customer features such as age, frequent flyer status, annual income class, services opted, account synchronization to social media, and hotel booking status, along with corresponding churn labels.
Thyroid	The Thyroid dataset contains patient features such as age, gender, smoking history, thyroid function, physical examination findings, pathology, and tumor staging information, along with corresponding recurrence outcomes.
Heloc	The Heloc dataset contains credit risk features such as trade information, external risk estimates, and payment history, along with corresponding risk performance labels (Bad or Good).
Consumer Behavior	The Consumer Behavior dataset contains customer demographic information such as age, gender, income, spending score, education level, marital status, children, and location, along with corresponding product categories (food or home).
Health Metrics	The Health Metrics dataset contains patient health indicators such as age, gender, height, weight, heart rate, blood pressure, and cholesterol levels, along with corresponding risk levels (low, medium, or high).
Real Estate	The Real Estate dataset contains property features such as area, location, age, renovation level, price, house type, and traffic convenience, along with corresponding school district indicators (yes or no).
Social Network	The Social Network dataset contains user social media features such as age, country, daily posts, following counts, followers counts, average likes, average likes from following, and account age exponent, along with corresponding influence levels (ranging from 0 to 3).

Table 10: Attributes in different datasets.

Dataset	Input Dim	Output Dim	MLP Architecture	Batch Size	LR	Optimizer	Epochs
Travel	6	2	Input → Attn → Block(64) → Block(32) → Block(16) → Linear(8) → Linear(2)	64	0.001	Adam ($\beta_1 = 0.5,\ \beta_2 = 0.9$)	100
Heloc	22	2	Input → Attn → Block(64) → Block(32) → Block(16) → Linear(8) → Linear(2)	64	0.001	Adam	100
Sick	27	2	Input → Attn → Block(64) → Block(32) → Block(16) → Linear(8) → Linear(2)	64	0.001	Adam	100
Thyroid	16	2	Input → Attn → Block(64) → Block(32) → Block(16) → Linear(8) → Linear(2)	64	0.001	Adam	100
Consumer Behavior	8	2	Input → Attn → Block(64) → Block(32) → Block(16) → Linear(8) → Linear(2)	64	0.001	Adam	100
Health Metrics	8	3	Input → Attn → Block(64) → Block(32) → Block(16) → Linear(8) → Linear(3)	64	0.001	Adam	100
Real Estate	7	2	Input → Attn → Block(64) → Block(32) → Block(16) → Linear(8) → Linear(2)	64	0.001	Adam	100
Social Network	8	4	Input → Attn → Block(64) → Block(32) → Block(16) → Linear(8) → Linear(4)	64	0.001	Adam	100

Table 11: Implementation details of the Core Set algorithm.

Detailed Explanations on the Evaluation Metrics. In this work, to be consistent with EPIC, we also adopt the weighted Macro-F1 Score, Balanced Accuracy (BAL ACC), Sensitivity, and Specificity as the

main evaluation metrics, which measure the overall imbalanced classification performance, the average of per-class recalls across all classes, and the corresponding recalls for the minority and majority classes, respectively.

In classification metrics, sensitivity is essentially the recall of the positive (minority) class, measuring the model's accuracy on samples from the minority class. Conversely, specificity is the recall for the negative (majority) class, measuring the model's accuracy on majority-class samples. Balanced accuracy is defined as the average of per-class recall across all classes (majority and minority).

The F1 score is the harmonic mean of precision and recall. In multi-class scenarios, the weighted Macro-F1 Score is computed by: (1) calculating per-class F1 scores using a one-vs-rest approach (treating each class as positive and all others as negative), then (2) taking the weighted average of these F1 scores, where weights are determined by each class's support (number of samples).

Implementation of Core Set. In Table 11, we give the implementation details and training configurations of the Core Set algorithm. We set K to be *100* for core set selection when choosing the Top-K most representative samples for each class.

A.5 Prompt Design

Stage	Prompt Name	Main Purpose	Input Information	Output Information
Initialization	MetaData Description	Inject domain knowledge and establish category label semantics	*Domain definitions:* Class labels (hypothyroidism), patient demographics (age, sex), laboratory values (TSH, T3, TT4, T4U, FTI)	Contextual grounding for LLMs (no direct output)
Prompt 1	Relationship Analysis	Guide model to explore variable interactions and correlations	*Analysis directive:* core set and the MetaData description as domain knowledge	Statistical/semantic associations (e.g., "High TSH correlates with hypothyroidism")
Prompt 2	Constraint Derivation	Extract rules and constraints from prior analysis	*Prior analysis results* + directive to define generation rules and constraints	Qualitative and quantitative rules for structured generation
Prompt 3	Data Generation	Generate structured synthetic data using derived constraints	*Derived constraints* + class balance requirements	Structured synthetic samples with balanced class distribution

Table 12: Overview of the three-stage prompt design for synthetic data generation. Each stage progressively builds upon the previous one to establish domain knowledge, analyze relationships, and generate structured synthetic data.

In Table 12, we give a sketch of our three-stages prompting strategy:

Initialization. As with EPIC, in the initial step, we construct a contextualized metadata description with background explanations to define and briefly describe key variables (e.g., TSH, T3, FTI), thereby infusing domain knowledge into the language model. Furthermore, categorical samples are annotated with alphanumeric identifiers (e.g., A, B, C, D) to enhance discriminative awareness of class labels. This prompt serves as the contextual basis.

(i) Relationship Analysis Prompt Phase. This stage employs carefully designed prompts to guide the LLMs in analyzing relationships between features and target variables, leveraging both knowledge and statistical analysis perspectives. By inputting small-scale core-set samples, the model identifies potential correlations and interaction terms, thereby establishing plausible inter-attribute relationships that can be used to generate subsequent data.

(ii) Constraint Derivation Prompt Phase. Following the initial attribute relationship analysis, we use these findings as contextual input to guide the model in refining and generating data construction rules under constraints. These rules encompass qualitative or quantitative inter-attribute relationships (e.g., "TSH is positively correlated with T3"), thereby providing explicit constraint specifications for the generative model.

(iii) Data Generation Prompt Phase. In the final stage, we leverage the constraints derived in the preceding phase and iteratively feed batches of the dataset into the LLMs to synthesize more representative and balanced training data. Crucially, this phase incorporates a dynamic guidance mechanism in which each generated batch is immediately assessed for quality. The evaluation results are transformed into

Prompt	Output
Prompt 1	Please analyze the relationships between these features and the churn (Churn) class. Identify any significant correlations or patterns that could help predict customer churn. Identify potential interactions among these features that may provide insights into customer behavior and churn likelihood.
Output1(analysis_results)	From the given data, it appears that there are some potential relationships between the features and customer churn. 1. Age: In some cases, younger customers are more likely to churn compared to older customers...
Prompt 2	{analysis_results}. Based on the above background data and the relationships among the data, rules and constraints for data generation are established.
Output2(constraints)	Rule 1: Customers who are in their 20s and opt for more services are more likely to churn compared to older customers who opt for fewer services. Rule 2: Frequent flyers are less likely to churn compared to customers who do not frequently fly. Rule 3:...
Prompt 3_1	{constraints}, ensure the class generation is balanced.
Prompt 3_2	You are generating tabular data. Here is the quality evaluation report: 1. Mean and Standard Deviation Differences: - Age: Mean diff = 0.03, Std dev diff = 0.30....
output of Prompt 3(Prompt 3_1 + Prompt 3_2)	Churn Age FrequentFlyer AnnualIncomeClass ServicesOpted AccountSyncedToSocialMedia BookedHotelOrNot A. IHU,30.0,YMP,CL2,4.0,NXU,U0X IHU,31.0,YBW,OI8,2.0,NXU,EUA B. HRL,29.0,YMP,T6L,2.0,NXU,EUA HRL,31.0,YBW,CL2,4.0,NXU,U0X

Table 13: Prompt Examples on the Travel Dataset.

Prompt	Output
Prompt 1	Please analyze the relationships between these features and the recurrence of thyroid cancer (Recurred), identifying any significant correlations or patterns that could help predict cancer recurrence and potential interactions among features.
Output1(analysis_results)	From the given data, we can identify potential patterns and correlations that could help predict cancer recurrence: 1. Age: Older age may be correlated with a higher likelihood of cancer recurrence. ...
Prompt 2	{analysis_results}. Based on the above background data and the relationships among the data, rules and constraints for data generation are established.
Output2(constraints)	Rules and constraints for data generation based on the relationships between the features and the recurrence of thyroid cancer could include: 1. Age must be recorded accurately and consistently, as older age may be correlated with a higher likelihood of cancer recurrence...
Prompt 3_1	{constraints}, Ensure the class generation is balanced.
Prompt 3_2	You are generating tabular data. Here is the quality evaluation report: 1. Mean and Standard Deviation Differences: - Age: Mean diff = 1.09, Std dev diff = 2.80....
Output of Prompt 3(Prompt 3_1 + Prompt 3_2)	Recurred, Age, Gender, Smoking, Hx Smoking, Hx Radiotherapy, Thyroid Function, Physical Examination, Adenopathy, Pathology, Focality, Risk, T, N, M, Stage, Response A. A8O,39,A6I,GQP,Z2Y,BFG,BMN,KMR,P1R, VDC,IOU,EOT,B8U,OLC,QA8,WY1,I8L A8O,26,LPT,GQP,Z2Y,BFG,HLJ,KMR,P1R, VDC,UE4,EOT,B8U,T47,QA8,WY1,I8L B. N5Q,53,LPT,W6O,Z2Y,BFG,BMN,MQ8,P1R, VDC,UE4,HGR,B8U,T47,QA8,WY1,GC4 N5Q,35,A6I,GQP,Z2Y,BFG,BMN,MQ8,P1R, VDC,IOU,EOT,B8U,OLC,QA8,WY1,LSU

Table 14: Prompt Examples on the Thyroid Dataset.

structured guidance prompts that direct subsequent generation iterations, ensuring continuous quality improvement and adherence to the established constraints. This process ensures both usability and diversity of the generated data, thereby enhancing performance for imbalanced classification tasks.
Finally, Tables 13 and 14 provide the concrete prompts used in RDDG on the Travel and Thyroid datasets, respectively.

A.6 Comparisons Between EPIC and RDDG

As LLM-based in-context learning approaches for tabular data synthesis, both EPIC and RDDG are model-training-free and highly efficient, yielding high-quality generated data. Both approaches adopt batch-wise synthetic data generation pipelines. Their main differences include: i) RDDG is a progressive framework with CoT steps, which break the in-context learning-based generation process into relation mining, data generation, and constraint optimization. In contrast, EPIC lacks such a learning design. ii) RDDG devises the self-reinforcing feedback mechanism that makes automatic assessment on the quality of the generated data in the preceding round with respect to the reference set (a batch of real data), and such evaluation results are turned into feedback prompts and incorporated in the subsequent in-context learning-based generation step. However, EPIC lacks such a feedback mechanism. iii) We formulate the self-reinforcing feedback process as a Bayesian calibration problem and establish theoretical guarantees for our framework. Specifically, we prove the Bayes-optimal performance of our approach and show that, under certain assumptions, our feedback mechanism converges to these optimal strategies. Moreover, extensive experiments demonstrate that RDDG achieves significantly better performance than EPIC in both imbalanced classification and data fidelity.

A.7 Pseudo Code of RDDG

Algorithm 1 CreateFeedback: Self-Reinforcing Feedback Mechanism

Require: Quality metrics Q_{stat}, Q_{corr}, Q_{dist}
Initialize feedback: $\mathcal{F} = \{\}$
if Q_{stat}.mean_diff $> \tau_{mean}$ **then**
  $\mathcal{F} \leftarrow \mathcal{F} \cup \{$"Adjust mean values closer to: " + target_means$\}$
end if
if Q_{stat}.std_diff $> \tau_{std}$ **then**
  $\mathcal{F} \leftarrow \mathcal{F} \cup \{$"Maintain variance similar to: " + target_stds$\}$
end if
if Q_{corr}.max_diff $> \tau_{corr}$ **then**
  Identify problematic attribute pairs (a_i, a_j)
  $\mathcal{F} \leftarrow \mathcal{F} \cup \{$"Strengthen correlation between " + $(a_i, a_j)\}$
end if
if Q_{dist}.ks_statistic $> \tau_{ks}$ **then**
  Identify distribution deviations
  $\mathcal{F} \leftarrow \mathcal{F} \cup \{$"Align distribution patterns for: " + attributes$\}$
end if
Format feedback as structured prompt guidance
return $\mathcal{F}$

Algorithm 2 RDDG: Relational Data Generator with Dynamic Guidance

Require: Training dataset $\mathcal{D}_{train} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$, Target synthetic samples N_{target}, Reference set size B, Core set size per class K, LLM model $\mathcal{M}$

Ensure: Synthetic dataset $\mathcal{D}_{syn}$

Core Set Construction
Initialize MLP classifier f_θ
Partition training into phases: $\mathcal{T} = \{\mathcal{T}_{early}, \mathcal{T}_{mid}, \mathcal{T}_{late}\}$
for each training phase $t \in \mathcal{T}$ **do**
  for each sample $(\mathbf{x}_i, y_i) \in \mathcal{D}_{train}$ **do**
    Compute L2 error: $e_i^t = \|\mathbf{y}_{pred} - \mathbf{y}_{true}\|_2^2$
  end for
end for
for each sample i **do**
  Compute variance: $Var_i = \sum_{t \in \mathcal{T}} \text{Var}(e_i^t)$
end for
for each class $c \in \mathcal{C}$ **do**
  $\mathcal{S}_c = \text{argtop}_K(\{Var_i \mid y_i = c\})$ {Select top-K variance samples}
  if $|\mathcal{S}_c| < K$ **then**
    Apply replacement sampling to reach K samples
  end if
end for
$\mathcal{D}_{core} = \bigcup_{c \in \mathcal{C}} \mathcal{S}_c$ {Combine coresets}

Phase 1: Relationship Analysis
Construct metadata prompt: $P_{meta} = \text{DescribeAttributes}(\mathcal{D}_{train})$
Initialize LLM with context: $\mathcal{M}.\text{init}(P_{meta})$
Generate relationship analysis prompt: $P_{rel} = \text{BuildRelationshipPrompt}(\mathcal{D}_{core})$
Extract relationships: $\mathcal{R} = \mathcal{M}.\text{analyze}(P_{rel}, \mathcal{D}_{core})$

Phase 2: Constraint Derivation
Generate constraint prompt: $P_{const} = \text{BuildConstraintPrompt}(\mathcal{R})$
Derive constraints: $\mathcal{C}_{rules} = \mathcal{M}.\text{extract\_constraints}(P_{const}, \mathcal{R})$

Phase 3: Batch-wise Data Generation with Dynamic Guide
Initialize synthetic dataset: $\mathcal{D}_{syn} = \emptyset$, $i \leftarrow 1$
Initialize feedback: $\mathcal{F}_0 = \emptyset$
Partition original data into batches: $\mathcal{B} = \{\mathcal{B}_1, \mathcal{B}_2, ..., \mathcal{B}_m\}$ where $|\mathcal{B}_j| = B$
while $|\mathcal{D}_{syn}| < N_{target}$ **do**
  Generate Batch:
  Build generation prompt: $P_{gen} = \text{BuildGenPrompt}(\mathcal{B}_i, \mathcal{C}_{rules}, \mathcal{F}_{i-1})$
  Generate samples: $\mathcal{S}_i = \mathcal{M}.\text{generate}(P_{gen})$
  Quality Evaluation:
  Compute statistical consistency: $Q_{stat} = \text{CompareStats}(\mathcal{S}_i, \mathcal{B}_i)$
  Compute correlation preservation: $Q_{corr} = \text{PearsonDiff}(\mathcal{S}_i, \mathcal{B}_i)$
  Compute distribution consistency: $Q_{dist} = \text{KSTest}(\mathcal{S}_i, \mathcal{B}_i)$
  Self-reinforcing Feedback Update:
  Generate feedback: $\mathcal{F}_i = \text{CreateFeedback}(Q_{stat}, Q_{corr}, Q_{dist})$
  Update Dataset:
  $\mathcal{D}_{syn} \leftarrow \mathcal{D}_{syn} \cup \mathcal{S}_i$
  $i \leftarrow (i \mod m) + 1$ {Cycle through batches}
end while
return $\mathcal{D}_{syn}$