TABDLM: Free-Form Tabular Data Generation via Joint Numerical–Language Diffusion

Donghong Cai[1] **Jiarui Feng**[1] **Yanbo Wang**[2] **Da Zheng**[3]
Yixin Chen[1] **Muhan Zhang**[2]
{cai.d, feng.jiarui, ychen25}@wustl.edu, wangyanbo@stu.pku.edu.cn
zhengda.zheng@antgroup.com, muhan@pku.edu.cn
[1]Washington University in St. Louis [2]Peking University [3]Ant Group

Abstract

Synthetic tabular data generation has attracted growing attention due to its importance for data augmentation, foundation models, and privacy. However, real-world tabular datasets increasingly contain *free-form text* fields (e.g., reviews or clinical notes) alongside structured numerical and categorical attributes. Generating such heterogeneous tables with joint modeling of **different modalities** remains challenging. Existing approaches broadly fall into two categories: diffusion-based methods and LLM-based methods. Diffusion models can capture complex dependencies over numerical and categorical features in continuous or discrete spaces, but extending them to open-ended text is nontrivial, since they model each column as a continuous value or one of a closed set of categories, whereas text lives in an exponentially large sequence space. In contrast, LLM-based generators naturally produce fluent text, yet their discrete tokenization can distort precise or wide-range numerical values, hindering accurate modeling of both numbers and language. In this work, we propose TABDLM, a unified framework for free-form tabular data generation via a joint numerical–language diffusion model built on masked diffusion language models (MDLMs). TABDLM models textual and categorical features through masked diffusion, while modeling numerical features with a continuous diffusion process through learned specialized numeric tokens embedding; bidirectional attention then captures cross-modality interactions within a single model. Extensive experiments on diverse benchmarks demonstrate the effectiveness of TABDLM compared to strong diffusion- and LLM-based baselines. Our code is available at https://github.com/ilikevegetable/TabDLM.

1 Introduction

Tabular data is one of the most fundamental data types in modern machine learning and is indispensable across domains such as finance [37], healthcare [17], and social media [24]. In practice, however, deploying machine learning on tabular datasets faces recurring obstacles, including privacy and security constraints [12, 2], limited data availability [9], and missing values [50, 47]. These challenges motivate synthetic tabular data generation, which seeks to produce synthetic samples that match the schema and key statistical properties of the original dataset for replacing the original data.

High-fidelity synthetic tabular data generation remains challenging because real-world tables exhibit complex dependencies across columns [45, 5] and often contain a mixture of numerical, categorical, and free-form textual features [33]. Existing approaches can broadly be categorized into diffusion-based and large language model (LLM)-based methods. Diffusion-based methods employ diffusion processes [40, 13, 39, 3] to learn denoising transformations that approximate the data distribution during training and to generate realistic samples from noise during inference. Such models have been widely applied to tabular data generation, particularly for continuous (numerical) features [20,

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

19, 50]. More recent work has explored combining continuous and discrete diffusion to jointly model numerical and categorical attributes [49, 23, 25, 38]. However, extending standard diffusion models to open-ended textual fields remains challenging due to the **exponential size of the text space**. Existing tabular diffusion models [23, 49, 38] decode every non-numerical cell into one of a fixed set of categories, so they cannot generate open-vocabulary text. In contrast, LLM-based methods naturally support free-form text generation [5, 51] owing to their strong language modeling capabilities. However, their token-level representations can be unreliable for modeling high-precision or wide-range numerical values [46], as **numbers are often fragmented into multiple tokens**. Furthermore, **autoregressive generation enforces a left-to-right dependency structure** that is misaligned with the mutually dependent relationships across tabular columns.

To address the above limitations, we propose TABDLM, a unified framework that integrates the complementary strengths of diffusion-based and LLM-based modeling. To preserve the most effective modeling paradigm for each modality, TABDLM employs continuous diffusion for numerical columns and language models for categorical and free-form textual columns. However, rather than relying on autoregressive language models, we adopt Masked Diffusion Language Models (MDLMs) as the backbone architecture. MDLMs offer two key advantages. First, their diffusion-based formulation enables TABDLM to model numerical and language modalities within a unified generative process. Second, unlike autoregressive models, MDLMs employ bidirectional attention, enabling the model to capture mutual dependencies among tabular columns. To enable joint numerical–language diffusion within a single MDLM, we introduce a trainable numerical tokenization module into the MDLM architecture, which allows continuous diffusion to represent each numerical value with a single token. As a result, TABDLM learns to **jointly denoise numerical values and language content** during training and generates synthetic tabular data by simultaneously performing continuous diffusion and masked language diffusion during inference. The overview of TABDLM is shown in Figure 1. We evaluate TABDLM across a range of scenarios and benchmarks. On tables with free-form text, which specialized tabular diffusion models cannot generate, TABDLM outperforms all language-model baselines. On conventional structured tables, it outperforms language-model baselines and remains competitive with specialized tabular diffusion models.

2 Methods

2.1 Preliminary

Problem setup and notation. Let $\mathcal{D} = \{\mathbf{x}^{(i)}\}_{i=1}^{N}$ denote a tabular dataset with M columns. Each record is $\mathbf{x}^{(i)} = (x_1^{(i)}, \ldots, x_M^{(i)})$, where columns can be either numerical, categorical, or textual. We use index sets $\mathcal{I}_{\text{num}}$, $\mathcal{I}_{\text{cat}}$, and $\mathcal{I}_{\text{text}}$ to denote numerical, categorical, and text columns, respectively. Our goal is to learn a generative model $p_\theta(\mathbf{x})$ that can sample realistic records while capturing cross-column dependencies.

Continuous diffusion. We model the distribution of a continuous vector $\mathbf{z}_0 \in \mathbb{R}^d$ using a diffusion process defined by the stochastic differential equation (SDE) $\mathrm{d}\mathbf{z} = \mathbf{f}(\mathbf{z}, t)\mathrm{d}t + g(t)\mathrm{d}\mathbf{w}$. In the variance-exploding (VE) formulation [41], we set $\mathbf{f}(\mathbf{z}, t) = \mathbf{0}$ and $g(t) = \sqrt{2\dot{\sigma}(t)\sigma(t)}$, where $\sigma(t) : [0, 1] \to \mathbb{R}_+$ is a strictly increasing function governing the noise level and $\dot{\sigma}(t) = \frac{\mathrm{d}\sigma(t)}{\mathrm{d}t}$ denotes its time derivative. From a probabilistic perspective, this SDE corresponds to a continuous-time limit of a Markov chain where Gaussian noise is progressively added to the data. The transition kernel $q(\mathbf{z}_t \mid \mathbf{z}_s)$ for $s < t$ and the marginal $q(\mathbf{z}_t \mid \mathbf{z}_0)$ are given by:

$$\begin{aligned} q(\mathbf{z}_t \mid \mathbf{z}_s) &= \mathcal{N}\big(\mathbf{z}_s,\, (\sigma^2(t) - \sigma^2(s))\mathbf{I}\big)\,, \\ q(\mathbf{z}_t \mid \mathbf{z}_0) &= \mathcal{N}\big(\mathbf{z}_0,\, \sigma^2(t)\mathbf{I}\big)\,. \end{aligned} \tag{1}$$

The reverse generative process solves the probability flow ODE [41], as follows:

$$\mathrm{d}\mathbf{z} = -\frac{1}{2}g(t)^2\nabla_{\mathbf{z}} \log p_t(\mathbf{z})\mathrm{d}t = -\dot{\sigma}(t)\sigma(t)\nabla_{\mathbf{z}} \log p_t(\mathbf{z})\mathrm{d}t. \tag{2}$$

We use a parameterized denoising model $\boldsymbol{\epsilon}_\theta(\mathbf{z}_t, t)$ that estimates the score function via $\nabla_{\mathbf{z}} \log p_t(\mathbf{z}) \approx -\boldsymbol{\epsilon}_\theta(\mathbf{z}_t, t)/\sigma(t)$. Training minimizes the denoising error:

$$\mathcal{L}_{\text{diff}} = \mathbb{E}_{t,\mathbf{z}_0,\boldsymbol{\epsilon}}\left[\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\mathbf{z}_0 + \sigma(t)\boldsymbol{\epsilon}, t)\|_2^2\right], \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \tag{3}$$

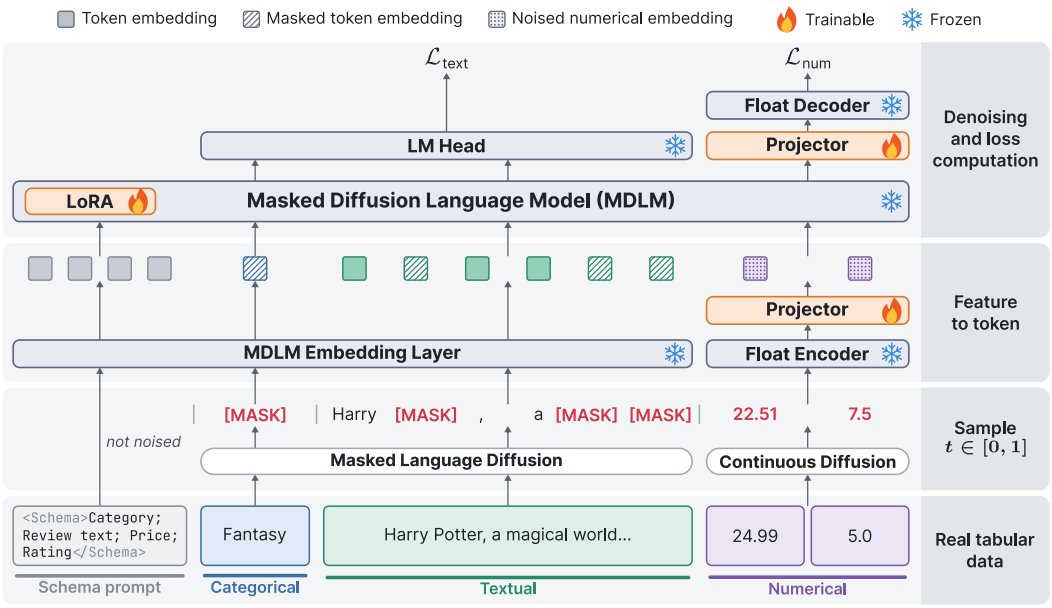


Figure 1: Overview of TABDLM. During training, given an input tabular sample, TABDLM applies masked language diffusion to categorical and free-form textual features, and continuous diffusion to numerical features. Noisy inputs are mapped to token embeddings via a textual embedding layer and a numerical encoder. An MDLM then denoises these embeddings to reconstruct the original sample. Training updates only the projector in the numerical encoder and decoder, as well as the LoRA modules within each MDLM layer.

Masked diffusion language models (MDLMs). MDLMs [35, 28] can be viewed as a discrete-state diffusion process [3] over token sequences. Let $\mathbf{s}_0 = (s_1, \ldots, s_L)$ be a length-L sequence with tokens in a vocabulary $\mathcal{V}$ augmented with an absorbing mask state $m = [\text{MASK}]$. Using a token-wise Markov chain, the forward noising kernel factorizes as

$$q(\mathbf{s}_t \mid \mathbf{s}_{t-1}) = \prod_{j=1}^{L} q(s_{t,j} \mid s_{t-1,j}), \tag{4}$$

with a masking schedule $\beta_t \in [0, 1]$ and absorbing transitions

$$q(s_{t,j} \mid s_{t-1,j}) = \begin{cases} 1, & s_{t-1,j} = m,\ s_{t,j} = m, \\ \beta_t, & s_{t-1,j} \neq m,\ s_{t,j} = m, \\ 1 - \beta_t, & s_{t-1,j} \neq m,\ s_{t,j} = s_{t-1,j}, \\ 0, & \text{otherwise}, \end{cases} \tag{5}$$

Equivalently, the marginal has the closed form $q(s_{t,j} = s_{0,j} \mid s_{0,j}) = \bar{\alpha}_t$ and $q(s_{t,j} = m \mid s_{0,j}) = 1 - \bar{\alpha}_t$, where $\bar{\alpha}_t = \prod_{\tau=1}^{t}(1 - \beta_\tau)$. The reverse model uses a bidirectional Transformer to predict the original token distribution from the partially masked sequence:

$$p_\theta(\mathbf{s}_0 \mid \mathbf{s}_t) = \prod_{j=1}^{L} p_\theta(s_{0,j} \mid \mathbf{s}_t), \tag{6}$$

and is typically trained by maximizing the log-likelihood of the ground-truth tokens at corrupted (masked) positions:

$$\mathcal{L}_{\text{MDLM}} = \mathbb{E}_{t, \mathbf{s}_0, \mathbf{s}_t} \left[- \sum_{\{j:\, s_{t,j} = [\text{MASK}]\}} \log p_\theta(s_{0,j} \mid \mathbf{s}_t) \right]. \tag{7}$$

At sampling time, MDLMs start from an all-mask sequence and iteratively unmask tokens according to p_θ.

2.2 The Model Architecture of TABDLM

We first present the architecture of TABDLM. As illustrated in Figure 1, TABDLM comprises five main components: (1) a numerical encoder module, (2) a text embedding layer, (3) a masked diffusion language model, (4) a numerical decoder module, and (5) an LM head. We describe each component in detail below. Note that in this section we use $\hat{x}_j^{(i)}$ instead of $x_j^{(i)}$ to denote the noisy version of the input. We will describe how we obtain the $\hat{x}_j^{(i)}$ from $x_j^{(i)}$ in Section 2.3.

Number encoder module. Given an input numerical value $\hat{x}_j^{(i)} \in \mathbb{R}, j \in \mathcal{I}_{\text{num}}$, the number encoder module transforms the number into a token embedding with the same size as the MDLM. To achieve this, we first use quantile normalization [4] to standardize the numerical values as done in previous works [43, 38]. Then, we use a pretrained Multi-Layer Perceptron (MLP) as the float number encoder to convert the normalized values into an r-dimensional embedding vector. Finally, a trainable projection is used to align the r-dimensional embedding vector into the d-dimensional MDLM embedding space:

$$\hat{\mathbf{z}}_j^{(i)} = \text{ENC}(\hat{x}_j^{(i)}), \;\; \mathbf{z}_j^{(i)} = \text{PROJ}_e(\hat{\mathbf{z}}_j^{(i)}), \;\; j \in \mathcal{I}_{\text{num}}, \tag{8}$$

where ENC and PROJ_e denote the MLP-based encoder and projection module, respectively, with $\hat{\mathbf{z}}_j^{(i)} \in \mathbb{R}^r$ and $\mathbf{z}_j^{(i)} \in \mathbb{R}^d$. In our implementation, we pretrained ENC based on Griffin [43] and keep it fixed throughout all experiments, while PROJ_e is jointly optimized during training.

Text embedding layer. The text embedding layer transforms textual input features $\hat{x}_j^{(i)}, j \in \mathcal{I}_{\text{cat}} \cup \mathcal{I}_{\text{text}}$ into a token embedding sequence. We directly adopt the pretrained embedding table from the underlying MDLM. Formally,

$$\mathbf{z}_j^{(i)} = (\mathbf{z}_{j,1}^{(i)}, \ldots, \mathbf{z}_{j,l_j^i}^{(i)}) = \text{EMB}(\hat{x}_j^{(i)}), j \in \mathcal{I}_{\text{cat}} \cup \mathcal{I}_{\text{text}}, \tag{9}$$

where each token embedding $\mathbf{z}_{j,k}^{(i)} \in \mathbb{R}^d, k \in [l_j^i]$ and l_j^i denotes the token sequence length of column j in sample i, as each textual column may be mapped to multiple tokens in the MDLM embedding space.

Masked diffusion language model. The masked diffusion language model is employed to jointly denoise both numerical and textual tokens. Formally, given an input sequence of token embeddings $\mathbf{z}^{(i)} = (\mathbf{z}_1^{(i)}, \ldots, \mathbf{z}_S^{(i)})$ of length S and a diffusion time step t, the MDLM outputs a denoised sequence $\mathbf{o}^{(i)}$ that estimates the corresponding representations at time $t = 0$. In TABDLM, the input token sequence is constructed by concatenating four components: a schema prompt, categorical features, textual features, and numerical features. The schema prompt provides a textual description of the feature type and the semantic meaning of each column, which is fixed for a given dataset. In detail, the input token sequence for sample i is represented as:

$$\mathbf{z}^{(i)} = \big(\mathbf{z}_p, (\mathbf{z}_j^{(i)} | j \in \mathcal{I}_{\text{cat}} \cup \mathcal{I}_{\text{text}}), (\mathbf{z}_j^{(i)} | j \in \mathcal{I}_{\text{num}})\big), \tag{10}$$

where $\mathbf{z}_p$ is the token embedding sequence for the schema prompt and $\mathbf{z}_j^{(i)} = (\mathbf{z}_{j,1}^{(i)}, \ldots, \mathbf{z}_{j,l_j^i}^{(i)})$ for any $j \in \mathcal{I}_{\text{cat}} \cup \mathcal{I}_{\text{text}}$. The forward process of MDLM can be described as:

$$\mathbf{o}^{(i)} = \text{MDLM}(\mathbf{z}^{(i)}, t). \tag{11}$$

For the MDLM architecture, we adopt a standard transformer architecture with bidirectional attention [42, 7]. For numerical tokens, an additional positional embedding is added to encode the diffusion noise level applied to the input, following the design in DiT [31]. No such embedding is added for textual tokens, as the noise level can be implicitly captured by the proportion of masked tokens.

Number decoder module. The number decoder module is used to decode the output token embedding for the numerical value back to the original number. We use a symmetric version of the number encoder module with one trainable projection, along with a pretrained MLP decoder:

$$\bar{\mathbf{o}}_j^{(i)} = \text{PROJ}_d(\mathbf{o}_j^{(i)}), \;\; \bar{x}_j^{(i)} = \text{DEC}(\bar{\mathbf{o}}_j^{(i)}), \;\; j \in \mathcal{I}_{\text{num}}, \tag{12}$$

where DEC and PROJ_d denote the MLP-based decoder and projection module, respectively, with $\mathbf{o}_j^{(i)} \in \mathbb{R}^d$ representing the output token embedding from the MDLM and $\bar{\mathbf{o}}_j^{(i)} \in \mathbb{R}^r$. Similarly, the DEC is pretrained jointly with the float number encoder and fixed during the experiment.

LM head. Finally, the LM head is used to decode the output token embedding for the textual value back to the original text. We directly use the frozen LM head from MDLM:

$$\bar{x}_j^{(i)} = \text{HEAD}(\mathbf{o}_j^{(i)}), \quad j \in \mathcal{I}_{\text{cat}} \cup \mathcal{I}_{\text{text}}. \tag{13}$$

2.3 The Forward Process of TABDLM

The forward diffusion process typically adds noise to the input. In TABDLM, this process is applied to both numerical and textual modalities; for clarity, we describe the forward process for each modality separately.

Forward process of numerical features. For numerical features, we adopt continuous diffusion to better capture fine-grained distributions. Given a clean input value $x_j^{(i)}, j \in \mathcal{I}_{\text{num}}$, we first sample a time $t \in [0, 1]$. Then, according to Equation 1, the forward process adds noise as follows:

$$\hat{x}_j^{(i)} = x_j^{(i)} + \sigma(t)\epsilon, \;\; \epsilon \sim \mathcal{N}(0, 1), \;\; j \in \mathcal{I}_{\text{num}}. \tag{14}$$

Note that the noise is added to the normalized value.

Forward process of textual features. For textual features, we adopt discrete masked diffusion, as used in MDLMs. Given a text sequence $x_j^{(i)} = (x_{j,1}^{(i)}, \ldots, x_{j,l_j^i}^{(i)}), j \in \mathcal{I}_{\text{cat}} \cup \mathcal{I}_{\text{text}}$, we first sample a time $t \in [0, 1]$. Then, according to Equation 5 and the accompanying discussion, the forward process independently transforms each token into a mask token with probability $1 - \bar{\alpha}_t$, resulting in the masked token sequence $\hat{x}_j^{(i)} = (\hat{x}_{j,1}^{(i)}, \ldots, \hat{x}_{j,l_j^i}^{(i)})$.

Finally, in TABDLM, we use a single t for both continuous diffusion and discrete masked diffusion. This design enables the model to jointly denoise numerical and textual modalities under a unified noise level.

2.4 Training of TABDLM

During training, we optimize TABDLM to jointly denoise both the numerical modality and textual modality. Specifically, given an output sequence $\mathbf{o}^{(i)} = \big(\mathbf{o}_p, (\mathbf{o}_j^{(i)} | j \in \mathcal{I}_{\text{cat}} \cup \mathcal{I}_{\text{text}}), (\mathbf{o}_j^{(i)} | j \in \mathcal{I}_{\text{num}})\big)$ from MDLM, we first transform the numerical token embedding back to real numbers using the number decoder module and then the loss is computed separately for continuous diffusion and masked diffusion:

$$\mathcal{L}_{\text{num}} = \frac{1}{N} \sum_{i=1}^{N} \sum_{j \in \mathcal{I}_{\text{num}}} \left\| x_j^{(i)} - \bar{x}_j^{(i)} \right\|_2^2, \quad j \in \mathcal{I}_{\text{num}}, \tag{15}$$

$$\begin{aligned} \mathcal{L}_{\text{text}} = & \frac{1}{N} \sum_{i=1}^{N} \sum_{\hat{x}_{j,k}^{(i)} = \text{[MASK]}} \text{CE} < \bar{x}_{j,k}^{(i)}, x_{j,k}^{(i)} >, \\ & k \in [l_j^i], \quad j \in \mathcal{I}_{\text{cat}} \cup \mathcal{I}_{\text{text}}, \end{aligned} \tag{16}$$

where $\text{CE} < \cdot, \cdot >$ is the cross entropy loss. The final objective is optimized using a step-dependent weighting schedule:

$$\mathcal{L} = \mathcal{L}_{\text{text}} + \lambda(s)\, \mathcal{L}_{\text{num}}, \;\; \lambda(s) = \lambda_{\max} \cdot \min\left(1, \frac{s}{s_{\text{warm}}}\right), \tag{17}$$

where s denotes the global optimization step, and this warm-up schedule stabilizes the alignment of newly initialized numerical projections with the pretrained MDLM backbone. We use default hyperparameters $\lambda_{\max} = 1$ and $s_{\text{warm}} = 2000$. In practice, we freeze the pretrained weights of the MDLM and introduce trainable LoRA modules [14] into both the feed-forward and attention components of each transformer layer.

2.5 The backward sampling in TABDLM

Finally, we describe the sampling process of TABDLM. After training, generation starts from noise and progressively denoises the inputs to produce synthetic tabular samples following the learned data distribution. For numerical modality, the process is initialized with pure Gaussian noise, while for textual modality, it starts from a fully masked token sequence. We then iteratively apply Equation 2 and Equation 6 to the numerical and textual modalities to recover clean data. More details are provided in Appendix B.

3 Related Works

Existing methods in tabular data generation can be classified into three categories based on their underlying frameworks.

VAE/GAN-based tabular generators. This line of work formulates tabular data generation using Variational Autoencoders (VAEs) [22] or Generative Adversarial Networks (GANs) [10]. Representative methods include CTGAN and TVAE [45]. GOGGLE [26] further enhances this paradigm by explicitly modeling column dependencies through a Graph Neural Network–augmented VAE architecture. However, these approaches often lack sufficient expressivity when confronted with complex tabular distributions involving intricate feature interactions.

Diffusion-based tabular generators. Motivated by the strong generative capacity of diffusion models [13], a growing body of work adapts diffusion processes for tabular data generation. Early methods model numerical and categorical features using separate discrete-time diffusion [3], as in Kotelnikov et al. [23], Lee et al. [25], but discretization can lead to looser ELBO bounds and suboptimal generation quality [41, 21]. More recent approaches encode tabular features into continuous latent spaces and apply Gaussian diffusion [50, 49]. However, such latent modeling introduces additional encoding overhead and may only indirectly capture heterogeneous feature interactions. Recently, CDTD [27] explores feature-wise noise schedules under continuous diffusion, and TabDiff [38] further extends it to mixed-type feature-level diffusion. Diffuse Everything [34] couples continuous and masked discrete diffusion through decoupled per-modality noise times and is applied to text–image generation and mixed-type tabular data generation; its tabular instantiation draws every categorical column from a fixed category set, with no free-form text column. TabNAT [48], designed specifically for tables, pairs continuous diffusion for numerical features with a bidirectional masked Transformer over column-specific categorical indices, so its discrete generation is likewise restricted to fixed category sets. In contrast, TABDLM leverages a pretrained MDLM with a large semantic vocabulary to jointly generate numerical, categorical, and free-form textual fields within a single model.

Language model-based tabular generators. Recent advances in large language models have also inspired language model–based approaches for tabular data generation. These methods serialize each row into a text sequence and fine-tune an autoregressive language model to capture row-level distributions, as exemplified by GReaT [5] with a GPT-2 [32] backbone. DiffLM [51] integrates VAEs and latent diffusion within a language modeling framework. However, DiffLM decodes each row, numbers included, with a frozen autoregressive LLM, so numerical values are generated as digit tokens. TabMT [11] and TabularMDLM [15] generate rows by masked token prediction, but likewise represent every value, numbers included, as discrete tokens. To the best of our knowledge, TABDLM is the first method for tabular data in which one pretrained masked diffusion language model denoises numerical values by continuous diffusion and categorical values and free-form text by token-level masked diffusion.

4 Experiments

In this section, we conduct extensive experiments to evaluate the proposed TABDLM. Specifically, we aim to answer the following questions. **Q1:** Can TABDLM effectively model the mutual dependencies among numerical, categorical, and text columns? **Q2:** How well does TABDLM perform on real-world tabular data generation tasks involving free-form text columns? **Q3:** Can TABDLM achieve performance on par with existing methods on tasks that do not include textual columns? We

implement TABDLM based on LLaDA-8B [28] and TABDLM$_{170M}$ based on MDLM [35] (170M parameters) to examine whether our framework remains effective with a much smaller backbone. Other implementation details can be found in Appendix B.

4.1 Results on synthetic tabular datasets

In this section, we answer the question **Q1** through two carefully designed synthetic tabular datasets.

Datasets. Existing real-world tabular generation datasets predominantly focus on numerical and categorical features and lack text fields. Thus, we construct two synthetic tabular datasets whose text column is rendered from the numerical and categorical columns: **MathExpr** and **ProfileBio**. Each MathExpr sample includes two floating-point variables, categorical features indicating the unary and binary operators applied to them, and a textual column containing the corresponding LaTeX expression. ProfileBio contains numerical and categorical attributes describing a person's demographic and educational background, along with a textual biography generated from the other columns. For both datasets, accurate generation requires models to **capture correlations between numerical, categorical, and textual columns**. Examples are shown in Tables 5 and 6, and the generation process in Appendix A.1. Both datasets serve as controlled diagnostics: the text field is rendered from the structured fields by a rule never shown to the models, so the consistency between the text and the structured fields can be checked exactly for every generated row, which real free-form text does not allow. Open-ended text generation is evaluated on the real-world datasets in Section 4.2.

Baselines. Given these two datasets contain text columns, which specialized tabular diffusion models cannot generate, we compare the proposed TABDLM with autoregressive and masked diffusion LLMs. Specifically, we include Qwen2.5 (7B/14B) with 3-shot in-context learning (ICL), and Qwen2.5-7B and LLaDA-8B with LoRA-based supervised fine-tuning (SFT). We further include GReaT [5], which fully fine-tunes GPT-2, and DiffLM [51], which trains a latent diffusion model on top of a VAE with a frozen Mistral-7B [16] decoder.

Metrics. We evaluate generated data along two groups. **1) Fidelity:** Shape and Trend measure marginal column-wise similarity and pairwise dependencies among numerical and categorical columns, both reported as error rates; **2) Cross-field consistency:** for MathExpr, we report Operation Match Rate (Op-MR) and Expression Match Rate (Exp-MR), measuring whether operators and numerical literals in the generated LaTeX align with the structured fields. For ProfileBio, we report Biography Match Rate (Bio-MR), checking consistency between the generated biography and structured attributes. Detailed definitions are provided in Appendix A.2.

Table 1: Evaluation results on the **MathExpr** and **ProfileBio** datasets (%).

Method	MathExpr				ProfileBio		
	Shape ↓	Trend ↓	Op-MR ↑	Exp-MR ↑	Shape ↓	Trend ↓	Bio-MR ↑
GReaT	19.28	25.40	97.67	93.07	**2.76**	10.90	97.89
DiffLM	6.84	17.70	97.31	97.12	18.82	31.18	68.74
Qwen2.5-7B$_{ICL}$	39.04	56.65	53.93	53.92	41.29	64.52	7.97
Qwen2.5-14B$_{ICL}$	25.61	36.07	76.01	75.12	43.97	65.00	13.00
Qwen2.5-7B$_{SFT}$	5.07	50.81	99.68	99.34	10.51	36.26	98.51
LLaDA-8B$_{SFT}$	5.41	42.54	98.36	97.90	5.81	44.14	97.63
TABDLM$_{170M}$	**2.12**	**4.42**	99.00	96.92	5.90	9.41	86.59
TABDLM	3.69	6.07	**99.99**	**99.80**	4.84	**7.81**	**98.60**

Results. As shown in Table 1, our method achieves the lowest Trend error and the highest match rates on both MathExpr and ProfileBio. On MathExpr, both variants of our model obtain lower Shape and Trend errors than every baseline: compared with the strongest baseline on each metric, TABDLM reduces them by 27.2% and 65.7%. Since MathExpr requires the generated LaTeX expression to be consistent with both numerical values and categorical operator columns, these results indicate that TABDLM can preserve explicit dependencies across numerical, categorical, and textual fields rather than merely matching marginal distributions. On ProfileBio, the highest Match Rate shows that TABDLM generates biographies that remain semantically aligned with the corresponding structured attributes. Together, the strong fidelity and cross-field consistency results provide a direct answer to **Q1**: TABDLM can effectively model mutual dependencies among heterogeneous tabular fields

through joint numerical-language denoising. TABDLM$_{170M}$ recovers the structured dependencies of both datasets, whereas its 12.0-point lower Bio-MR on ProfileBio suggests that the 8B backbone helps keep the generated text consistent with the structured fields.

4.2 Results on real-world tabular datasets with free-form text features

In this section, we evaluate TABDLM on real-world tabular datasets with free-text to answer **Q2**.

Dataset and Baselines. We use two real-world datasets, Amazon and Arxiv, both derived from the RelBench [33] relational benchmark. Amazon merges several `rel-amazon` tables into one table with numerical, categorical, and long-text fields (e.g., title, description, and review); Arxiv is a paper-level table from `rel-arxiv` with numerical, categorical, and free-form text fields (e.g., title, arxiv code, and abstract). Details are in Appendix A.1, and the baselines are the same as those in Section 4.1.

Table 2: Results on the **Amazon** and **Arxiv** datasets. (* indicates *(w/o nums)*)

Method	Amazon				Arxiv			
	Shape (%)↓	Trend (%)↓	MLE* ↑	MLE ↑	Shape (%)↓	Trend (%)↓	MLE* ↑	MLE ↑
Real	0.0	0.0	.905	.906	0.0	0.0	.968	.968
GReaT	6.91	5.94	.874	.877	6.72	6.43	.599	.603
DiffLM	17.23	43.84	.819	.817	15.17	16.38	.848	.837
Qwen2.5-7B$_{ICL}$	37.93	68.27	.636	.629	68.04	42.34	.521	.516
Qwen2.5-14B$_{ICL}$	39.19	78.22	.684	.679	40.79	34.35	.556	.527
Qwen2.5-7B$_{SFT}$	11.41	46.31	.824	.834	11.19	10.02	.701	.727
LLaDA-8B$_{SFT}$	7.76	37.13	.891	.892	14.10	43.81	.931	.930
TABDLM$_{170M}$	9.15	7.48	.861	.864	10.15	8.05	.944	**.951**
TABDLM	**4.67**	**5.33**	**.893**	**.895**	**6.30**	**6.30**	**.946**	.948

Metrics. Similar to Section 4.1, we evaluate **1) Fidelity:** Shape and Trend. Additionally, we include **2) Downstream utility:** Machine Learning Efficiency (MLE), which measures how well predictive models trained on synthetic data generalize to real test data. Finally, to evaluate cross-field consistency between free-form text and the remaining numerical and categorical features, we follow an MLE-like protocol: we first serialize all non-numerical fields into structured text and encode them with the sentence embedding model Nomic [29], then train an XGBoost Classifier [6] using either (i) text embeddings only or (ii) text embeddings concatenated with numerical features, which isolates the contribution of generated text/categorical fields. MLE is reported as AUC.

Results. The results on Amazon and Arxiv are presented in Table 2. Our method achieves the best performance across Shape, Trend, and MLE on both datasets. Compared with LLaDA-8B$_{SFT}$, which shares the backbone and the training data of TABDLM but represents numerical values as digit tokens, TABDLM reduces the Shape and Trend errors by 39.8–85.6% on both tables. GReaT is the strongest baseline on Shape and Trend but has a substantially lower downstream utility on Arxiv. These results show that TABDLM effectively preserves structured-field marginal distributions and pairwise structured dependencies in real-world datasets that also contain free-form textual fields. The distribution of every numerical column is compared with the real data in Appendix C.6.

Regarding downstream utility, the MLE of our method closely approaches the real-data upper bound. This indicates that the generated text, categorical, and numerical fields jointly preserve label-relevant information. Using only the generated text and categorical fields (MLE*), TABDLM attains the highest score on both tables, within .012 and .022 of real data, so the generated text itself carries the label-relevant information. To test whether the generated numbers agree with the generated text, we compare the word count of each generated Arxiv `abstract` with the generated `abstract_length` of its row (Figure 2): the share of rows within $\pm 10\%$ is 39.0 percentage points higher for TABDLM than for the strongest baseline, so the numbers and text generated by TABDLM are consistent with each other. We evaluate the generated text and present a case study in Appendices C.7 and C.8.

Comparing our two variants, TABDLM reduces the Shape and Trend errors of TABDLM$_{170M}$ by 21.7–49.0% and, on Amazon, generates text with higher judged consistency and fewer hallucinations (Appendix C.7), so a stronger backbone brings further gains. TABDLM$_{170M}$ nevertheless remains strong overall despite its much smaller backbone. Together, these results provide a direct answer to

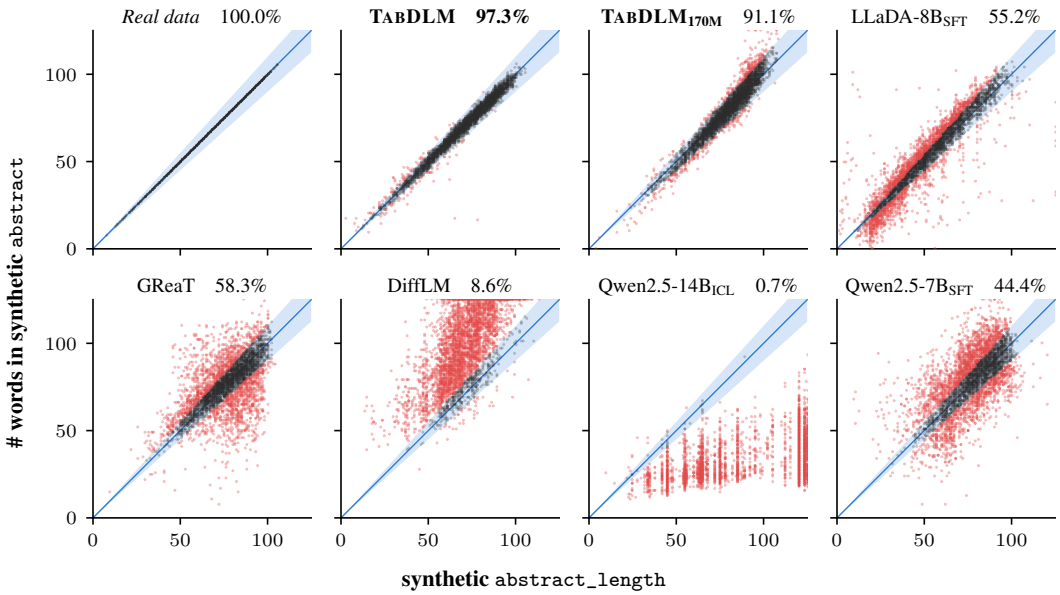


Figure 2: Number–text consistency on Arxiv. Each point is a synthetic row: its `abstract_length` (x) against the word count of its `abstract` (y). Each panel reports the percentage of rows whose word count is within $\pm 10\%$ of `abstract_length` (shaded band); red points fall outside the band.

Q2: TabDLM performs strongly on real-world tabular generation tasks involving free-form text, preserving both structured-field fidelity and downstream task-relevant information across modalities.

4.3 Results on real-world tabular datasets without free-form text features

Finally, we answer **Q3** on traditional tabular data generation benchmarks without free-form text features, enabling comparison with existing specialized methods that cannot generate such text.

Datasets, baselines, and metrics. We conduct experiments on five real-world tabular datasets: Adult, Default, Shoppers, Magic, and Beijing. Detailed dataset profiles are presented in Appendix A.1.5. Baselines are widely-used tabular generators from four categories: 1) GAN-based: CTGAN [45]; 2) VAE-based: TVAE [45] and GOGGLE [26]; 3) Autoregressive Tabular Language Model: GReaT [5] and DiffLM [51]; 4) Diffusion-based: STaSy [19], CoDi [25], TabDDPM [23], TabSyn [49], TabDiff [38], and TabNAT [48]. We evaluate along three groups of metrics, defined in Appendix A.2: **1) Fidelity:** Shape, Trend, α-Precision, and C2ST; **2) Downstream utility:** Machine Learning Efficiency (MLE); **3) Privacy:** Distance to Closest Record (DCR) evaluates privacy risk by measuring the distance between each synthetic sample and its nearest training sample, where overly small distances may indicate memorization. We report Shape, Trend in the main paper and include α-Precision, C2ST, MLE, and DCR results in Appendix C.

Results. As reported in Table 3, TabDLM performs competitively with state-of-the-art tabular diffusion models on Shape and Trend. **This is notable because such tabular-specific synthetic methods operate in a relatively restricted feature space** (e.g., categorical columns are represented via one-hot vectors or limited discrete states), whereas TabDLM is built to jointly model heterogeneous fields and ultimately support open-ended text generation, yet it still remains strong on purely numerical and categorical tabular datasets. TabDLM$_{170M}$ also performs strongly with a $47\times$ smaller backbone, which offers an efficient option when no free text is involved (Appendix B). Beyond this, TabDLM consistently and substantially outperforms all language-based baselines, with average gains of 62.1% in Shape and 84.8% in Trend. Notably, **language-based generators often achieve reasonable Shape but markedly worse Trend**. A likely reason is that row serialization simplifies matching column-wise marginals, as each field can be generated locally to fit its token-level distribution. However, Trend depends on modeling **joint** dependencies among numerical and categorical columns, which is particularly difficult to preserve when numbers are serialized as

Table 3: Results on the error rates (%) of **Shape ↓ / Trend ↓**. Each cell reports Shape / Trend. Full per-metric results are provided in Appendix C.1. (**B**: best overall; **B**: best in language-based model)

Method	Adult	Default	Shoppers	Magic	Beijing	Average
CTGAN	16.84 / 20.23	16.83 / 26.95	21.15 / 13.08	9.81 / 7.00	21.39 / 22.95	17.20 / 18.04
TVAE	14.22 / 14.15	10.17 / 19.50	24.51 / 18.67	8.25 / 5.82	19.16 / 18.01	15.26 / 15.23
GOGGLE	16.97 / 45.29	17.02 / 21.94	22.33 / 23.90	1.90 / 9.47	16.93 / 45.94	15.03 / 29.31
STaSy	11.29 / 14.51	5.77 / 5.96	9.37 / 8.49	6.29 / 6.61	6.71 / 8.00	7.89 / 8.71
CoDi	21.38 / 22.49	15.77 / 68.41	31.84 / 17.78	11.56 / 6.53	16.94 / 7.07	19.50 / 24.46
TabDDPM	1.75 / 3.01	1.57 / 4.89	2.72 / 6.61	1.01 / 1.70	1.30 / 2.71	1.67 / 3.78
TabSyn	0.81 / 1.93	1.01 / 2.81	1.44 / 2.13	1.03 / 0.88	1.26 / 3.13	1.11 / 2.18
TabDiff	**0.63** / **1.49**	1.24 / 2.55	1.28 / 1.74	0.78 / **0.76**	1.03 / 2.59	0.99 / **1.83**
TabNAT	0.73 / 1.61	**0.80** / **2.21**	**1.11** / **1.67**	**0.62** / 1.37	**0.79** / **2.48**	**0.81** / 1.87
GReaT	12.12 / 17.59	19.94 / 70.02	14.51 / 45.16	16.16 / 10.23	8.25 / 59.60	14.20 / 40.52
DiffLM	9.74 / –	9.06 / –	10.07 / –	7.53 / –	6.35 / –	8.55 / –
Qwen2.5-7B$_{\text{ICL}}$	27.80 / 37.61	22.39 / 31.02	34.37 / 34.77	13.41 / 20.62	31.83 / 40.83	25.96 / 32.97
Qwen2.5-14B$_{\text{ICL}}$	25.17 / 42.67	21.52 / 28.72	51.21 / 42.32	17.51 / 18.11	29.76 / 38.20	29.03 / 34.00
Qwen2.5-7B$_{\text{SFT}}$	5.11 / 17.03	2.58 / 16.99	8.76 / 10.28	7.53 / 14.45	7.49 / 28.68	6.29 / 17.49
LLaDA-8B$_{\text{SFT}}$	1.69 / 26.15	2.00 / 26.30	13.56 / 16.42	5.60 / 13.71	3.64 / 27.90	5.30 / 22.10
TabDLM$_{\text{170M}}$	1.49 / **2.56**	3.64 / 6.19	4.54 / 4.62	**1.96** / **1.76**	**2.39** / **2.71**	2.80 / 3.57
TabDLM	**1.46** / 2.74	**1.18** / **2.33**	**2.05** / **2.40**	2.93 / 2.85	2.42 / 2.93	**2.01** / **2.65**

multi-token strings. In practice, numerical values are represented as subword token sequences with precision-sensitive semantics, whereas categorical fields act as discrete identifiers, making consistent numerical–categorical correlations hard to preserve under token-level generation. In contrast, TabDLM jointly generates numerical and categorical features within a unified diffusion process and leverages bidirectional interactions across columns, enabling more direct modeling of tabular dependencies than token-serialized row generation and yielding greater fidelity on structured datasets.

4.4 Ablation Study

We assess the contribution of each component of TabDLM with three variants: (i) TabDLM-noFloatAE, which replaces the pretrained float encoder/decoder with randomly initialized counterparts; (ii) TabDLM-onlyContDiff, which removes MDLM-based modeling for categorical fields; and (iii) TabDLM-onlyMDLM, which removes the continuous diffusion branch and processes numerical features as serialized tokens. Detailed variant descriptions are provided in Appendix C.9.

Table 4: Ablation study on the average performance across Adult, Default, and Shoppers.

Method	Shape ↓	Trend ↓	MLE ↑	C2ST ↑	α-Precision ↑
TabDLM-noFloatAE	2.15	4.34	.870	0.9256	98.50
TabDLM-onlyContDiff	4.19	7.28	.850	0.8399	91.70
TabDLM-onlyMDLM	5.75	22.96	.852	0.7892	88.58
TabDLM	**1.56**	**2.49**	**.874**	**0.9571**	**98.56**

The full TabDLM consistently outperforms all three variants. The largest drop occurs in TabDLM-onlyMDLM, supporting our motivation in Section 1 that subword tokenization fragments precision-sensitive numerical values and hampers fine-grained numerical–categorical correlation modeling. TabDLM-noFloatAE and TabDLM-onlyContDiff also degrade noticeably, validating the necessity of pretrained numerical encoding and MDLM-based modeling for categorical fields, respectively.

5 Conclusion

In this work, we propose TabDLM, to the best of our knowledge the first joint numerical–language diffusion method for generating tables with numerical, categorical, and free-form text columns. TabDLM leverages an MDLM with special numerical tokenization to allow a single model to denoise numbers by continuous diffusion and categories and text by masked diffusion. Extensive experiments validate the effectiveness of TabDLM over baseline models. Its generated numbers and text agree with each other, and its free-form text is both diverse and consistent with the other columns. We discuss limitations and future directions, including sampling efficiency and modality-specific noise schedules, in Appendix D.

Acknowledgments and Disclosure of Funding

Donghong Cai, Jiarui Feng, and Yixin Chen are supported by NSF grant CBET 2225809, and a Washington University Transcend Initiative grant. Yanbo Wang and Muhan Zhang are supported by the National Natural Science Foundation of China (62550138, 62276003). The authors declare no competing interests.

References

[1] Ahmed Alaa, Boris Van Breugel, Evgeny S Saveliev, and Mihaela Van Der Schaar. 2022. How faithful is your synthetic data? sample-level metrics for evaluating and auditing generative models. In *International conference on machine learning*. PMLR, 290–306.

[2] Samuel A Assefa, Danial Dervovic, Mahmoud Mahfouz, Robert E Tillman, Prashant Reddy, and Manuela Veloso. 2020. Generating synthetic data in finance: opportunities, challenges and pitfalls. In *Proceedings of the First ACM International Conference on AI in Finance*. 1–8.

[3] Jacob Austin, Daniel D Johnson, Jonathan Ho, Daniel Tarlow, and Rianne Van Den Berg. 2021. Structured denoising diffusion models in discrete state-spaces. *Advances in neural information processing systems* 34 (2021), 17981–17993.

[4] Benjamin M Bolstad, Rafael A Irizarry, Magnus Åstrand, and Terence P. Speed. 2003. A comparison of normalization methods for high density oligonucleotide array data based on variance and bias. *Bioinformatics* 19, 2 (2003), 185–193.

[5] 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*.

[6] Tianqi Chen and Carlos Guestrin. 2016. XGBoost: A Scalable Tree Boosting System. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2016).

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*. 4171–4186.

[8] Stefan Elfwing, Eiji Uchibe, and Kenji Doya. 2018. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *Neural networks* 107 (2018), 3–11.

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

[10] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. *Advances in neural information processing systems* 27 (2014).

[11] Manbir S. Gulati and Paul F. Roysdon. 2023. TabMT: Generating Tabular Data with Masked Transformers. In *Advances in Neural Information Processing Systems*.

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

[13] Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. *Advances in neural information processing systems* 33 (2020), 6840–6851.

[14] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. 2022. Lora: Low-rank adaptation of large language models. *ICLR* 1, 2 (2022), 3.

[15] Tokimasa Isomura, Ryotaro Shimizu, and Masayuki Goto. 2025. Generating realistic synthetic tabular data with integrated LLM and diffusion models. *Neurocomputing* 655 (2025), 131371.

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

[17] Alistair EW Johnson, Lucas Bulgarelli, Lu Shen, Alvin Gayles, Ayad Shammout, Steven Horng, Tom J Pollard, Sicheng Hao, Benjamin Moody, Brian Gow, et al. 2023. MIMIC-IV, a freely accessible electronic health record dataset. *Scientific data* 10, 1 (2023), 1.

[18] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. 2022. Elucidating the design space of diffusion-based generative models. *Advances in neural information processing systems* 35 (2022), 26565–26577.

[19] Jayoung Kim, Chaejeong Lee, and Noseong Park. 2023. STaSy: Score-based Tabular data Synthesis. In *The Eleventh International Conference on Learning Representations*.

[20] Jayoung Kim, Chaejeong Lee, Yehjin Shin, Sewon Park, Minjung Kim, Noseong Park, and Jihoon Cho. 2022. Sos: Score-based oversampling for tabular data. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*. 762–772.

[21] Diederik Kingma, Tim Salimans, Ben Poole, and Jonathan Ho. 2021. Variational diffusion models. *Advances in neural information processing systems* 34 (2021), 21696–21707.

[22] Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114* (2013).

[23] Akim Kotelnikov, Dmitry Baranchuk, Ivan Rubachev, and Artem Babenko. 2023. Tabddpm: Modelling tabular data with diffusion models. In *International conference on machine learning*. PMLR, 17564–17579.

[24] Himabindu Lakkaraju, Julian McAuley, and Jure Leskovec. 2013. What's in a name? understanding the interplay between titles, content, and communities in social media. In *Proceedings of the international AAAI conference on web and social media*, Vol. 7. 311–320.

[25] Chaejeong Lee, Jayoung Kim, and Noseong Park. 2023. Codi: Co-evolving contrastive diffusion models for mixed-type tabular synthesis. In *International Conference on Machine Learning*. PMLR, 18940–18956.

[26] Tennison Liu, Zhaozhi Qian, Jeroen Berrevoets, and Mihaela van der Schaar. 2023. Goggle: Generative modelling for tabular data by learning relational structure. In *The Eleventh International Conference on Learning Representations*.

[27] Markus Mueller, Kathrin Gruber, and Dennis Fok. 2023. Continuous diffusion for mixed-type tabular data. *arXiv preprint arXiv:2312.10431* (2023).

[28] Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, Jun Zhou, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. 2025. Large language diffusion models. *arXiv preprint arXiv:2502.09992* (2025).

[29] Zach Nussbaum, John Xavier Morris, Andriy Mulyar, and Brandon Duderstadt. 2025. Nomic Embed: Training a Reproducible Long Context Text Embedder. *Transactions on Machine Learning Research* (2025). Reproducibility Certification.

[30] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems* 32 (2019).

[31] William Peebles and Saining Xie. 2023. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*. 4195–4205.

[32] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog* 1, 8 (2019), 9.

[33] Joshua Robinson, Rishabh Ranjan, Weihua Hu, Kexin Huang, Jiaqi Han, Alejandro Dobles, Matthias Fey, Jan Eric Lenssen, Yiwen Yuan, Zecheng Zhang, Xinwei He, and Jure Leskovec. 2024. RelBench: A Benchmark for Deep Learning on Relational Databases. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*. `https://openreview.net/forum?id=WEFxOm3Aez`

[34] Kevin Rojas, Yuchen Zhu, Sichen Zhu, Felix X-F. Ye, and Molei Tao. 2025. Diffuse Everything: Multimodal Diffusion Models on Arbitrary State Spaces. In *Forty-second International Conference on Machine Learning*. `https://openreview.net/forum?id=AjbiIcRt6q`

[35] Subham Sekhar Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin T Chiu, Alexander Rush, and Volodymyr Kuleshov. 2024. Simple and Effective Masked Diffusion Language Models. In *Advances in Neural Information Processing Systems*, Vol. 37.

[36] Gerard Salton and Christopher Buckley. 1988. Term-weighting approaches in automatic text retrieval. *Information processing & management* 24, 5 (1988), 513–523.

[37] Timur Sattarov, Marco Schreyer, and Damian Borth. 2023. Findiff: Diffusion models for financial tabular data generation. In *Proceedings of the Fourth ACM International Conference on AI in Finance*. 64–72.

[38] Juntong Shi, Minkai Xu, Harper Hua, Hengrui Zhang, Stefano Ermon, and Jure Leskovec. 2025. TabDiff: a Mixed-type Diffusion Model for Tabular Data Generation. In *The Thirteenth International Conference on Learning Representations*.

[39] Jiaming Song, Chenlin Meng, and Stefano Ermon. 2020. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502* (2020).

[40] Yang Song and Stefano Ermon. 2019. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems* 32 (2019).

[41] 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*.

[42] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).

[43] Yanbo Wang, Xiyuan Wang, Quan Gan, Minjie Wang, Qibin Yang, David Wipf, and Muhan Zhang. 2025. Griffin: Towards a Graph-Centric Relational Database Foundation Model. *arXiv preprint arXiv:2505.05568* (2025).

[44] Chengyue Wu, Hao Zhang, Shuchen Xue, Zhijian Liu, Shizhe Diao, Ligeng Zhu, Ping Luo, Song Han, and Enze Xie. 2025. Fast-dllm: Training-free acceleration of diffusion llm by enabling kv cache and parallel decoding. *arXiv preprint arXiv:2505.22618* (2025).

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

[46] Haotong Yang, Yi Hu, Shijia Kang, Zhouchen Lin, and Muhan Zhang. 2024. Number cookbook: Number understanding of language models and how to improve it. *arXiv preprint arXiv:2411.03766* (2024).

[47] Jiaxuan You, Xiaobai Ma, Yi Ding, Mykel J Kochenderfer, and Jure Leskovec. 2020. Handling missing data with graph representation learning. *Advances in Neural Information Processing Systems* 33 (2020), 19075–19087.

[48] Hengrui Zhang, Liancheng Fang, Qitian Wu, and Philip S. Yu. 2025. TabNAT: A Continuous-Discrete Joint Generative Framework for Tabular Data. In *International Conference on Machine Learning*.

[49] 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*.

[50] S Zheng and N Charoenphakdee. 2022. Diffusion models for missing value imputation in tabular data. arXiv. *arXiv preprint arXiv:2210.17128* (2022).

[51] Ying Zhou, Xinyao Wang, Yulei Niu, Yaojie Shen, Lexin Tang, Fan Chen, Ben He, Le Sun, and Longyin Wen. 2025. Difflm: Controllable synthetic data generation via diffusion language models. In *Findings of the Association for Computational Linguistics: ACL 2025*. 20638–20658.

Contents of the Appendix

A Detailed Experiment Setups **16**

A.1 Datasets 16

A.2 Metrics 19

B Implementation Details **22**

C Additional Experimental Results **24**

C.1 Full results of Shape and Trend Metrics 24

C.2 Evaluation results of MLE for real-world tabular dataset 24

C.3 Evaluation results of α-Precision 24

C.4 Evaluation results of C2ST 25

C.5 Evaluation results of DCR 27

C.6 Distributions of the numerical columns 28

C.7 Quality of the generated free-form text 29

C.8 Case study 31

C.9 Description of Ablation Variants 33

D Limitations and Future Directions **33**

E Societal Impact **33**

A Detailed Experiment Setups

A.1 Datasets

A.1.1 MathExpr

MathExpr is a synthetic dataset designed to evaluate the joint generation of heterogeneous tabular records involving numerical values, categorical operators, and LaTeX expressions rendered from the structured columns. Each record contains two numerical columns, three categorical operator columns, and one text column:

$$\mathbf{r} = (x_1, x_2, o_1, o_2, o_3, e_{\texttt{latex}}).$$

Here, o_1 and o_2 are unary operators applied to x_1 and x_2, respectively, while o_3 is a binary operator that combines the two transformed terms. The text column $e_{\texttt{latex}}$ is a LaTeX expression deterministically constructed from the preceding structured columns.

We generate $10{,}000$ records and split them into a training/real set and a validation set with a ratio of $9{:}1$. During sampling, each model generates $9{,}000$ synthetic samples, and these synthetic samples are then used to compute the distribution fidelity metrics and expression consistency metrics described in Appendix A.2.6.

Numerical value sampling. We sample x_1 and x_2 from discrete supports with step size 0.1:

$$x_1 \in \{0.1, 0.2, \ldots, 6.0\}, \qquad x_2 \in \{3.0, 3.1, \ldots, 9.9\}.$$

To induce a non-uniform yet diverse distribution, we sample x_1 and x_2 from an equal-weight mixture of a discrete Gaussian distribution and a uniform distribution over the corresponding support. The discrete Gaussian components are centered at $\mu_1 = 3$ and $\mu_2 = 6.5$, respectively, with shared standard deviation $\sigma = 1.5$.

Categorical operator sampling. The unary operators o_1 and o_2 are sampled independently from

$$\mathcal{O}_{\text{unary}} = \{\texttt{none}, \texttt{log}, \texttt{exp}, \texttt{sqrt}, \texttt{sin}, \texttt{cos}, \texttt{tan}, \texttt{square}, \texttt{cube}\}.$$

Their categorical priors are:

$$p(o_1) = \{\texttt{none} : 0.18,\ \texttt{log} : 0.16,\ \texttt{sqrt} : 0.13,\ \texttt{square} : 0.12,\ \texttt{sin} : 0.10,\ \texttt{cos} : 0.10,\ \texttt{tan} : 0.07,\ \texttt{exp} : 0.07,\ \texttt{cube} : 0.07\},$$

$$p(o_2) = \{\texttt{none} : 0.22,\ \texttt{sin} : 0.14,\ \texttt{cos} : 0.14,\ \texttt{sqrt} : 0.12,\ \texttt{log} : 0.10,\ \texttt{square} : 0.09,\ \texttt{tan} : 0.07,\ \texttt{exp} : 0.06,\ \texttt{cube} : 0.06\}.$$

The binary operator o_3 is sampled from

$$\mathcal{O}_{\text{binary}} = \{\texttt{add}, \texttt{sub}, \texttt{mul}, \texttt{div}\},$$

with categorical prior

$$p(o_3) = \{\texttt{add} : 0.35,\ \texttt{mul} : 0.30,\ \texttt{sub} : 0.20,\ \texttt{div} : 0.15\}.$$

LaTeX expression construction. The expression string $e_{\texttt{latex}}$ is deterministically generated using a fixed grammar: unary operators are rendered as standard LaTeX commands (e.g., $\texttt{log} \mapsto \texttt{\textbackslash log}(\cdot)$, $\texttt{sqrt} \mapsto \texttt{\textbackslash sqrt}\{\cdot\}$, $\texttt{square} \mapsto (\cdot)^2$). Binary operators are rendered as $+$, $-$, $\texttt{\textbackslash times}$, or $\texttt{\textbackslash frac}\{\cdot\}\{\cdot\}$.

Table 5: An example sample from the **MathExpr** dataset.

Column	Example Value
x_1	2.70
x_2	6.40
operation_x1	`sin`
operation_x2	`log`
operation_between	`mul`
latex_expression	`\sin(2.70) \times \log(6.40)`

Table 6: An example sample from the **ProfileBio** dataset.

Column	Example Value
age	38
salary	135
sex	`female`
birth_state	`California`
college	`Harvard University`
degree	`master`
occupation	`software developer`
biography	`This female individual is in the career-building stage. She was born in California and completed higher education at Harvard University, earning a master degree. She works as a software developer. She earns a strong professional income.`

A.1.2 ProfileBio

ProfileBio is a synthetic dataset for evaluating the joint generation of mixed-type personal profiles with a biography text rendered from the structured columns. Each record contains two numerical columns, five categorical columns, and one text column:

$$\mathbf{r} = (\texttt{age}, \texttt{salary}, \texttt{sex}, \texttt{birth_state}, \texttt{college}, \texttt{degree}, \texttt{occupation}, \texttt{biography}).$$

The `biography` field is a natural-language paragraph deterministically constructed from the structured attributes.

Following the setup of MathExpr, we generate $10{,}000$ records and split them into a training/real set and a validation set with a ratio of $9{:}1$. During sampling, each model generates a synthetic dataset with the same cardinality as the real training set. We evaluate distribution fidelity and biography consistency using the metrics described in Appendix A.2.6.

Categorical value sampling. We sample `sex` uniformly from {`male`, `female`}. The attributes `birth_state` and `college` are sampled from fixed categorical priors over 10 states and 9 colleges, respectively. The `degree` attribute is sampled conditionally on `college`: for `stanford university` and `harvard university`, we use

$$p(\texttt{degree}) = (0.01, 0.29, 0.40, 0.30),$$

and for all other colleges, we use

$$p(\texttt{degree}) = (0.30, 0.50, 0.15, 0.05),$$

where the entries correspond to $(\texttt{associate}, \texttt{bachelor}, \texttt{master}, \texttt{doctoral})$. Finally, `occupation` is sampled from a degree-dependent categorical distribution. All occupation weights are initialized to 1. For `doctoral` degrees, the weights of `research specialist` and `education professional` are set to 6 and 4, respectively; for `associate` degrees, the weights of `customer services professional` and `construction professional` are set to 5 and 5, respectively. The weights are then normalized into a categorical distribution.

Numerical value sampling. We sample `age` uniformly from integers in $[21, 70]$, and draw `salary` from a Gaussian whose mean depends on `degree`, `occupation`, and `age`. The degree-dependent base mean is

$$\mu_0(\texttt{degree}) = \begin{cases} 82, & \texttt{associate}, \\ 125, & \texttt{bachelor or master}, \\ 178, & \texttt{doctoral}, \end{cases}$$

and, letting $\mathcal{O}^\star = \{\texttt{software developer}, \texttt{healthcare practitioner}\}$, the full mean is

$$\mu \;=\; \mu_0(\texttt{degree}) + 4 \cdot \mathbb{1}\{\texttt{occupation} \in \mathcal{O}^\star\} + 0.3\,(\texttt{age} - 45).$$

We then sample $\texttt{salary} \sim \mathcal{N}(\mu,\, 5^2)$, round it to the nearest integer, and clip it to $[75, 200]$.

Biography construction. The `biography` field is deterministically constructed from the structured attributes using a fixed template. The `age` and `salary` values are first mapped to coarse-grained textual descriptors, and Table 7 summarizes both the descriptor mappings and the biography template.

Table 7: **ProfileBio:** Age/salary mapping rules and the biography template.

Component	Rule / Template
Age descriptor	[21, 30]: `in the early career stage` [31, 40]: `in the career-building stage` [41, 50]: `in the established career stage` [51, 60]: `in the advanced career stage` [61, 70]: `in the late career stage`
Salary descriptor	[75, 100]: `a comfortable income` [101, 150]: `a strong professional income` [151, 200]: `a high-level income`
Biography template	`This {sex} individual is {age_desc}. {He/She} was born in {birth_state} and completed higher education at {college}, earning a {degree} degree. {He/She} works as a {occupation}. {He/She} earns {salary_desc}.`

Table 8: An example sample from the **Amazon** dataset.

Column	Example Value
price	5.98
review_time	1970 (days since earliest review date)
rating	5.0
verified	true
category	Science Fiction & Fantasy > Fantasy
brand	Visit Amazon's J. R. R. Tolkien Page
title	Hobbit
description	The enchanting prelude to "The Lord of the Rings"
review_text	My 13 yr. old grandson was very happy with this book. He likes to know the background of things and this helped him to understand this story.
summary	Grandson happy

Additional clarification. ProfileBio is a fully synthetic dataset constructed from predefined sampling rules and deterministic templates. It does not rely on, copy, or imitate any real individuals. Sensitive personal attributes such as race, religion, health status, or immigration background are intentionally excluded. While attributes such as `sex`, `birth_state`, and `occupation` are included to evaluate cross-field consistency, the salary generation process is explicitly designed to depend only on `age`, `degree`, and `occupation`, and does not condition on `sex` or `birth_state`. ProfileBio is intended solely as a controlled benchmark for evaluating joint generation fidelity and attribute–text consistency, rather than as a model of real-world socioeconomic distributions.

A.1.3 Amazon

Amazon is a real-world free-text tabular dataset constructed from the RelBench [33] `rel-amazon` relational benchmark, which contains linked product metadata and user reviews. We join the `review` and `product` tables on `product_id` and form a single mixed-type table with two numerical columns, two categorical columns, and six free-form text columns:

$$(\texttt{price}, \texttt{review_time}, \texttt{rating}, \texttt{verified}, \texttt{category}, \texttt{brand}, \texttt{title}, \texttt{description}, \texttt{review_text}, \texttt{summary}).$$

We sample $5{,}000$ examples in total and split them into a training/real set and a validation set with a ratio of 9:1. In addition, we sample another test set of $2{,}250$ examples for downstream utility evaluation. During sampling, each model generates $4{,}500$ synthetic examples, matching the cardinality of the real training set. We report distribution fidelity and downstream utility metrics as defined in Appendix A.2.6. Table 8 shows an example record.

A.1.4 Arxiv

Arxiv is another real-world free-text tabular dataset constructed from the RelBench `rel-arxiv` relational benchmark, which contains metadata and abstracts of arXiv papers. We construct a single

Table 9: An example record from the **Arxiv** dataset.

Column	Example Value
submission_time	1672 (days since earliest submission date in the dataset)
submission_year	2022
title_length	7
abstract_length	90
category	hep-ph
title	IRC-safe Graph Autoencoder for unsupervised anomaly detection
arxiv_code	arXiv:2204.12231
abstract	Anomaly detection through employing machine learning techniques has emerged as a novel powerful tool in the search for new physics beyond the Standard Model. Historically similar to the development of jet observables, theoretical consistency has not always assumed a central role in the fast development of algorithms and neural network architectures. In this work, we construct an infrared and collinear safe autoencoder based on graph neural networks by employing energy-weighted message passing. We demonstrate that whilst this approach has theoretically favourable properties, it also exhibits formidable sensitivity to non-QCD structures.

Table 10: Statistics of real-world tabular datasets. #Num denotes the number of numerical columns, and #Cat denotes the number of categorical columns. #Max Cat is the maximum number of categories among all categorical columns.

Dataset	#Rows	#Num	#Cat	#Max Cat	#Train	#Validation	#Test	Task
Adult	48,842	6	9	42	28,943	3,618	16,281	Classification
Default	30,000	14	11	11	24,000	3,000	3,000	Classification
Shoppers	12,330	10	8	20	9,864	1,233	1,233	Classification
Magic	19,019	10	1	2	15,215	1,902	1,902	Classification
Beijing	43,824	7	5	31	35,058	4,383	4,383	Regression

mixed-type table at the paper level with four numerical columns, one categorical column, and three free-form text columns:

$$(\texttt{submission_time}, \texttt{submission_year}, \texttt{title_length}, \texttt{abstract_length}, \texttt{category}, \texttt{title}, \texttt{arxiv_code}, \texttt{abstract}).$$

We sample $4{,}500$ examples for training and validation and split them into a real training set and a validation set with a ratio of 8:1. In addition, we sample another test set of $2{,}000$ examples for downstream utility evaluation. During sampling, each model generates a synthetic dataset with the same cardinality as the real training set. We report distribution fidelity and downstream utility metrics as defined in Appendix A.2.6. Table 9 shows an example record.

A.1.5 Real-world Tabular Datasets

We evaluate on five widely-used real-world tabular datasets from the UCI Machine Learning Repository.[1] These datasets cover both classification and regression tasks, providing a diverse testbed for regular tabular generation with only numerical and categorical features. Specifically, Adult, Default, Shoppers, and Magic are used for classification, while Beijing is used for regression. Dataset statistics and the corresponding train/validation/test splits are summarized in Table 10.

A.2 Metrics

A.2.1 Shape and Trend

We evaluate distribution fidelity on all datasets using two general-purpose metrics: **Shape** and **Trend**. Shape and Trend are adopted from SDMetrics[2], which quantify the marginal column-wise similarity and the pairwise dependency preservation between real and synthetic data, respectively.

[1] https://archive.ics.uci.edu/datasets

[2] https://docs.sdv.dev/sdmetrics

Shape. *Kolmogorov-Smirnov Test (KST):* This metric quantifies the alignment between the real distribution $p_r(x)$ and the synthetic distribution $p_s(x)$ by calculating the maximum divergence between their respective Cumulative Distribution Functions (CDFs):

$$\text{KST} = \sup_x |F_r(x) - F_s(x)|, \tag{18}$$

where $F_r(x)$ and $F_s(x)$ denote the CDFs derived from the probability densities:

$$F(x) = \int_{-\infty}^{x} p(t)\mathrm{d}t. \tag{19}$$

Total Variation Distance (TVD): For categorical variables, we evaluate the discrepancy in probability mass using the TVD. It is defined as half the sum of the absolute differences between the category frequencies observed in the real data, $R(\omega)$, and the synthetic data, $S(\omega)$:

$$\text{TVD} = \frac{1}{2}\sum_{\omega\in\Omega} |R(\omega) - S(\omega)|, \tag{20}$$

where Ω represents the set of all possible categories within a given column.

Trend. *Pearson Correlation Score:* We examine the preservation of linear dependencies between continuous columns using the Pearson correlation coefficient, $\rho_{x,y}$, defined as:

$$\rho_{x,y} = \frac{\text{Cov}(x,y)}{\sigma_x \sigma_y}, \tag{21}$$

where Cov represents covariance and σ denotes the standard deviation. To evaluate the overall preservation of these trends, we compute the Pearson Score as the normalized average absolute error between the correlation matrices of the real (ρ^R) and synthetic (ρ^S) datasets:

$$\text{Pearson Score} = \frac{1}{2}\mathbb{E}_{x,y}|\rho^R(x,y) - \rho^S(x,y)|. \tag{22}$$

Since $\rho \in [-1, 1]$, the factor of $1/2$ normalizes the score to the range $[0, 1]$, where a lower score indicates superior correlation preservation.

Contingency Similarity: To measure the consistency of pairwise associations between categorical columns A and B, we utilize a metric based on the Total Variation Distance applied to contingency tables. The Contingency Score is calculated as:

$$\text{Contingency Score} = \frac{1}{2}\sum_{\alpha\in A}\sum_{\beta\in B} |R_{\alpha,\beta} - S_{\alpha,\beta}|, \tag{23}$$

where $R_{\alpha,\beta}$ and $S_{\alpha,\beta}$ correspond to the joint frequencies of category pair (α, β) in the real and synthetic datasets, respectively.

A.2.2 α-Precision

Following Alaa et al. [1], α-Precision is a sample-level fidelity metric that measures whether each synthetic sample falls within the α-support of the real data distribution, i.e., the smallest region containing an α fraction of the real probability mass. Intuitively, it quantifies how typical synthetic samples are with respect to the real distribution rather than lying in low-density or out-of-distribution regions. A higher α-Precision indicates higher fidelity of the generated samples.

A.2.3 Detection Score

The Detection Score is computed via the Classifier Two-Sample Test (C2ST) implemented in SDMetrics, where a logistic regression classifier is trained to distinguish real samples from synthetic ones. The score is derived from the classifier's misclassification rate: when the synthetic distribution closely matches the real one, the classifier performs near chance level, yielding a higher Detection Score. We report the Detection Score in $[0, 1]$, with higher values indicating that synthetic samples are statistically indistinguishable from real samples.

A.2.4 Machine Learning Efficiency

For datasets with an associated downstream prediction task, we additionally report Machine Learning Efficiency (MLE) to assess utility by comparing the test performance of models trained on real versus synthetic samples. In particular, we apply MLE to the regular real-world tabular datasets. During evaluation, we train an XGBoost model on the real training set (further split with an 8:1 ratio for validation and hyperparameter tuning) and evaluate it on a held-out real test set. We then train an identical model on the synthetic dataset and evaluate it on the same real test set. We report the test performance of the model trained on synthetic data (AUC for classification and RMSE for regression); the closer it is to that of the model trained on real data, the better synthetic data can serve as a substitute for real data in downstream predictive modeling. The Average Gap in Table 15 is the relative drop in test performance when training on synthetic instead of real data, averaged over the datasets.

A.2.5 Distance Closest Record (DCR)

DCR evaluates whether the generative model memorizes training samples rather than learning the underlying distribution, serving as a data privacy metric. For each synthetic sample, we compute its distance to the nearest record in both the training set and a held-out test set, and report the proportion of synthetic samples whose nearest neighbor lies in the training set. A score close to 50% is ideal: it indicates that synthetic samples are equally close to training and test data, suggesting the model has captured the distribution rather than reproducing training instances.

A.2.6 Dataset-specific Metrics

MathExpr. Beyond standard distribution metrics (Shape and Trend), we evaluate whether the generated LaTeX expression is structurally and numerically consistent with the structured number and operation columns. We report: (1) Operation Match Rate (Op-MR), which verifies if the unary/binary operator tokens implied by the expression $e_{\texttt{latex}}$ strictly align with the structured operations (o_1, o_2, o_3); and (2) Expression match rate (Exp-MR), which evaluates the joint validity of structure and values. Specifically, a generated expression is considered a match only if it (i) satisfies operation correctness (as in Op-MR) and (ii) contains two numeric literals whose values align with the structured fields (x_1, x_2) up to a small relative tolerance. Concretely, letting $\hat{x}_1, \hat{x}_2$ be the literals extracted from the generated LaTeX string, we require $|\hat{x}_i - x_i|/x_i \leq \delta$ for $i \in \{1, 2\}$. We fix $\delta = 0.07$ to make the metric robust to minor numeric drift that can arise from stochastic generation and continuous-value approximation (e.g., 0.29 vs. 0.30), while still penalizing outputs whose numeric content is meaningfully inconsistent with the structured inputs.

ProfileBio. For ProfileBio, we assess cross-modality consistency between structured attributes and the generated biography text. We compute a rule-based Match Rate by checking whether the biography instantiates the required template slots with values consistent with the corresponding structured fields (see the example template in Table 7). Concretely, for each record we verify that the text reflects core attributes (e.g., `sex`, `birth_state`, `college`, `degree`, `occupation`) as well as the derived descriptors for continuous variables (`Age` and `Salary`). Since mapping continuous values into discrete natural-language descriptors introduces semantic fuzziness near bin boundaries, we apply a small boundary relaxation tolerance $\delta = 0.06$ when validating the age/salary descriptors. This avoids rigid thresholding artifacts where values close to a boundary may legitimately share descriptions from neighboring categories; for instance, age 30.5 can reasonably be described as either "in the early career stage" or "in the career-building stage."

Amazon. For the Amazon dataset, we measure downstream utility via an MLE-like protocol, with `rating` as the target label in $\{1, 2, 3, 4, 5\}$ (treated as a multi-class classification task). We first serialize all non-numerical fields into structured text and encode them using the sentence embedding model Nomic. We then train an XGBoost classifier using either (i) text embeddings only or (ii) text embeddings concatenated with numerical features, which helps isolate the contribution of generated text/categorical fields. We report Macro-AUC on the held-out test set as the utility metric.

Arxiv. Similar to the Amazon dataset, we measure downstream utility via an MLE-like protocol, with `Category` as the target label (treated as a multi-class classification task). The original `rel-arxiv` benchmark contains more than 50 paper categories, which would induce severe class imbalance and weaken the signal of cross-field consistency; we therefore restrict our benchmark to samples

from the 15 most frequent categories, yielding a relatively balanced 15-way classification task. We serialize all non-numerical fields into structured text and encode them using the sentence embedding model Nomic. We then train an XGBoost classifier using either (i) text embeddings only or (ii) text embeddings concatenated with numerical features. For each configuration, Macro-AUC on the held-out test is reported as the utility metric.

B Implementation Details

We implement TABDLM based on PyTorch [30]; the code is available at `https://github.com/ilikevegetable/TabDLM`. All experiments are run on a single NVIDIA A100-80GB GPU.

Data preprocessing. We follow the same preprocessing method as the prior tabular diffusion model TabDiff [38]. Missing numerical values are imputed with the column mean, and missing categorical values are treated as an additional category. To stabilize optimization across heterogeneous numerical scales, we apply a quantile-based transformation to numerical columns during training and invert the transform after sampling to recover values in the original space.

Data splits. We adopt the same data split protocol as the TabDiff setting: each dataset is partitioned into a real set and a test set. Models are trained on the real set. For downstream utility evaluation, we further split the real set into training and validation subsets and reserve the test set strictly for evaluation.

Architecture. We build TABDLM on top of the LLaDA-8B base model for all experiments. To incorporate numerical channels, each scalar value is first mapped into a d-dimensional latent using a lightweight pretrained float encoder, implemented as a 3-layer MLP with hidden width $\lfloor\sqrt{d}\rfloor$ and SiLU activations [8]. A float decoder (LayerNorm + linear projection) maps the latent back to a scalar. We set the numerical latent dimension to $d = 512$ by default and keep the pretrained float encoder/decoder frozen during joint training. To interface numerical latents with the MDLM embedding space, we further apply a two-stage projection: an *input projector* that maps per-feature latents from d to the MDLM hidden size D using a 2-layer MLP ($d \to 1024 \to D$) with SiLU and dropout (with LayerNorm), and an *output projector* that maps MDLM hidden states back to the numerical latent space via a symmetric 2-layer MLP ($D \to 1024 \to d$). The dimension D in LLaDA-8B is 4096.

Hyperparameter settings. We fine-tune models with LoRA. Unless otherwise stated, we use LoRA rank $r = 16$, scaling factor $\alpha = 32$, and dropout 0.05, and apply LoRA to the attention and MLP components of Transformer blocks. All models are trained and evaluated on a single NVIDIA A100-80GB GPU; training time, memory, and sampling cost are reported in Tables 11 and 12. Across methods, we keep the optimizer and training recipe identical; the only dataset-dependent choice is the number of training epochs due to varying dataset sizes. Specifically, we train for 10 epochs on Adult and Beijing, 30 epochs on Shoppers, 15 epochs on Magic and Default, and 75 epochs on MathExpr, Amazon, Arxiv, and ProfileBio. We use AdamW with learning rate 2×10^{-4}, warmup ratio 0.1, $(\beta_1, \beta_2) = (0.9, 0.98)$, weight decay 10^{-4}, and $\epsilon = 10^{-8}$. All experiments use the same fixed random seed and bf16 training when enabled.

Noise schedule. For the continuous numerical diffusion, we adopt the per-feature power-mean noise schedule following TabDiff [38], which extends the EDM parameterization [18] with a learnable shape parameter ρ_i for each numerical column. The schedule smoothly interpolates between $\sigma_{\min}$ and $\sigma_{\max}$ over normalized time $t \in [0, 1]$; we set $\sigma_{\min} = 0.002$ and $\sigma_{\max} = 80.0$ following EDM defaults. Formally, for each numerical feature $i \in \{1, \ldots, M_{\text{num}}\}$:

$$\sigma_{\rho_i}^{\text{num}}(t) = \left(\sigma_{\min}^{1/\rho_i} + t\left(\sigma_{\max}^{1/\rho_i} - \sigma_{\min}^{1/\rho_i}\right)\right)^{\rho_i}. \tag{24}$$

Learning ρ_i per column accommodates heterogeneous numerical distributions and yields a better-conditioned diffusion process than a single global ρ.

Sampling. TABDLM performs joint sampling over discrete tokens (categorical/text) and continuous numerical features through a coupled reverse process discretized into T steps; the full procedure is summarized in Algorithm 1. At each step, we (i) inject EDM-style churn noise [18] into the numerical state, (ii) embed both modalities into a shared MDLM input by mapping numerical values through the number encoder and tokens through the embedding table, (iii) run a single bidirectional MDLM forward pass that jointly conditions text denoising on noisy numericals and vice versa, and (iv) apply

Algorithm 1 Sampling

Require: Schema prompt $\mathbf{p}$; generation length G; number of reverse steps T; noise schedules $\{\sigma_t\}_{t=0}^{T}$ and churned schedule $\{\hat{\sigma}_t\}_{t=0}^{T}$; remasking policy $\pi \in \{\textsc{HighConf}, \textsc{Rand}\}$.
Ensure: Synthetic record $\mathbf{x} = \big(x_j\big)_{j \in \mathcal{I}_{\text{num}} \cup \mathcal{I}_{\text{cat}} \cup \mathcal{I}_{\text{text}}}$.

// Initialization
$\mathbf{z}_{\mathbf{p}} \leftarrow \text{EMB}(\mathbf{p})$
$\mathbf{x}_T^{\text{tok}} \leftarrow \texttt{[MASK]}^G$ ▷ discrete state for $\mathcal{I}_{\text{cat}} \cup \mathcal{I}_{\text{text}}$
$\mathbf{x}_T^{\text{num}} \sim \mathcal{N}\big(\mathbf{0},\, \sigma_{\max}^2 \mathbf{I}\big)$ ▷ initial state from a Gaussian prior
for $t = T, T-1, \ldots, 1$ **do**
  // (i) Numerical perturbation (EDM churn)
  $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$; $\quad \hat{\mathbf{x}}_t^{\text{num}} \leftarrow \mathbf{x}_t^{\text{num}} + \sqrt{\hat{\sigma}_t^2 - \sigma_t^2}\, \boldsymbol{\epsilon}$
  // (ii) Map both modalities to MDLM token embeddings
  $\mathbf{z}_t^{\text{num}} \leftarrow \text{PROJ}_e\big(\text{ENC}(\hat{\mathbf{x}}_t^{\text{num}})\big)$ ▷ Eq. (8)
  $\mathbf{z}_t^{\text{tok}} \leftarrow \text{EMB}(\mathbf{x}_t^{\text{tok}})$ ▷ Eq. (9)
  $\mathbf{z}_t \leftarrow \big[\mathbf{z}_{\mathbf{p}};\, \mathbf{z}_t^{\text{tok}};\, \mathbf{z}_t^{\text{num}}\big]$
  // (iii) Joint bidirectional denoising
  $\mathbf{o}_t = \big[\mathbf{o}_t^{\text{p}};\, \mathbf{o}_t^{\text{tok}};\, \mathbf{o}_t^{\text{num}}\big] \leftarrow \text{MDLM}(\mathbf{z}_t,\, t)$ ▷ Eq. (11)
  // (iv) Discrete reverse step: progressive unmasking
  $p_\theta(\cdot \mid \mathbf{x}_t^{\text{tok}}) \leftarrow \text{HEAD}(\mathbf{o}_t^{\text{tok}})$ ▷ Eq. (13)
  $\tilde{\mathbf{x}}_0^{\text{tok}} \sim p_\theta(\cdot \mid \mathbf{x}_t^{\text{tok}})$ ▷ sample candidate tokens
  $\mathbf{x}_{t-1}^{\text{tok}} \leftarrow \textsc{Unmask}_\pi\big(\mathbf{x}_t^{\text{tok}},\, \tilde{\mathbf{x}}_0^{\text{tok}},\, p_\theta,\, t\big)$ ▷ reveal G/T positions per step
  // (v) Continuous reverse step: EDM–Euler update
  $\tilde{\mathbf{x}}_t^{\text{num}} \leftarrow \text{DEC}\big(\text{PROJ}_d(\mathbf{o}_t^{\text{num}})\big)$ ▷ Eq. (12)
  $\mathbf{d}_t \leftarrow \big(\hat{\mathbf{x}}_t^{\text{num}} - \tilde{\mathbf{x}}_t^{\text{num}}\big)/\hat{\sigma}_t$
  $\mathbf{x}_{t-1}^{\text{num}} \leftarrow \hat{\mathbf{x}}_t^{\text{num}} + \big(\sigma_{t-1} - \hat{\sigma}_t\big)\, \mathbf{d}_t$ ▷ discretization of Eq. (2)
end for
// Finalization
$\big\{x_j\big\}_{j \in \mathcal{I}_{\text{cat}} \cup \mathcal{I}_{\text{text}}} \leftarrow \text{Detokenize}(\mathbf{x}_0^{\text{tok}})$
$\big\{x_j\big\}_{j \in \mathcal{I}_{\text{num}}} \leftarrow \text{Denorm}(\mathbf{x}_0^{\text{num}})$ ▷ numerical denormalization
return $\mathbf{x}$

modality-specific reverse updates: progressive unmasking on the discrete side and an EDM–Euler step on the continuous side.

For discrete unmasking we consider two policies: (*i*) **high-confidence** unmasking, which reveals positions with the highest predicted token probability, and (*ii*) **random** unmasking, which reveals a uniformly random subset. In both cases we reveal G/T tokens per step, ensuring a smooth transition from a fully-masked sequence to a fully-specified one. After the final step, we decode tokens through the LM head and denormalize the numerical values to obtain the final mixed-type sample.

TabDLM$_{170M}$. To test whether our formulation carries over to a different backbone and to measure the trade-off between text quality and efficiency, we instantiate the same framework on the 170M-parameter masked diffusion language model of Sahoo et al. [35], which is pretrained on OpenWebText with the GPT-2 tokenizer. The number encoder and decoder, the two projectors and the training objective are identical to those of TabDLM; besides the backbone, TabDLM$_{170M}$ differs in that, given its size, all parameters are fine-tuned instead of using LoRA (AdamW, learning rate 4×10^{-4}, batch size 64).

Computational cost. Table 11 reports the training and sampling cost on Shoppers, and Table 12 the sampling time per generated row on Amazon, all measured on a single NVIDIA A100-80GB GPU. Each of the T reverse steps (Algorithm 1) is a single forward pass of the backbone over the full sequence, i.e., the prompt, the G token positions, and one position per numerical value, and updates the text and the numerical values together. Generating a row therefore takes T forward passes, where T is a hyperparameter set independently of G, since G/T tokens are unmasked per step. Because each numerical value occupies a single position rather than a sequence of digit tokens, the sequence stays short, which keeps each forward pass cheap on wide tables. TabDLM is fine-tuned with LoRA

(r=16, α=32 on the attention and MLP blocks), i.e., 41.9M trainable parameters on top of the frozen 8B backbone. Specialized tabular diffusion models such as TabDiff are the cheapest to sample per generated row, but they cannot generate free-form text. TABDLM$_{170M}$ trains faster than TabDiff with less memory and samples faster than both Qwen2.5-7B baselines on Amazon, while staying close to TABDLM on average on the tables of Table 3; our formulation therefore largely does not depend on a large backbone for structured fidelity. TABDLM still improves on TABDLM$_{170M}$ in Shape and Trend on Amazon and Arxiv and in Bio-MR on ProfileBio (Tables 1 and 2), which is why we adopt LLaDA-8B by default. Finally, TABDLM and LLaDA-8B$_{SFT}$ cost the same at the same number of steps, so the sampling time of TABDLM is that of its backbone, and acceleration techniques for diffusion language models such as KV caching [44] are compatible with our formulation.

Table 11: Training and sampling cost on Shoppers. Memory is measured under each method's default training batch size (bs); the sampling time of TABDLM$_{170M}$ and TABDLM is per step at batch size 16.

Method	Backbone / trainable	Train time	Train memory	Sampling
TabDiff	10.6M / 10.6M	1.0 h	72 GB (bs 1000)	0.072 s/step $\times$ 50
TABDLM$_{170M}$	170M / 170M	0.77 h	20 GB (bs 64)	0.017 s/step $\times$ 48
TABDLM	8B / 41.9M (LoRA)	9.0 h	54 GB (bs 32)	0.39 s/step $\times$ 48

Table 12: Sampling time per generated row on Amazon. Specialized tabular diffusion models are absent because they cannot generate free-form text.

Method	Sampling steps	Time / row
GReaT (124M)	–	$\sim$ 0.05 s
Qwen2.5-7B$_{ICL}$	–	$\sim$ 0.22 s
Qwen2.5-14B$_{ICL}$	–	$\sim$ 0.34 s
Qwen2.5-7B$_{SFT}$	–	$\sim$ 0.38 s
LLaDA-8B$_{SFT}$	128	$\sim$ 3.8 s
TABDLM$_{170M}$	128	$\sim$ 0.17 s
TABDLM	128	$\sim$ 3.8 s

C Additional Experimental Results

C.1 Full results of Shape and Trend Metrics

Shape and Trend are fidelity metrics measuring column-wise marginal distributions and pair-wise column dependencies, respectively. Tables 13 and 14 report the complete results on real-world tabular datasets without free-form text features.

C.2 Evaluation results of MLE for real-world tabular dataset

In this section, we provide the evaluation results of MLE for real-world tabular datasets without free-form text features in Table 15. As shown in the table, TABDLM achieves competitive MLE performance on Adult, Default, Shoppers, and Magic, demonstrating its ability to faithfully capture the underlying data distribution and support high-quality synthetic data for downstream training. Notably, TABDLM attains the best overall score on Shoppers, and on Default even surpasses models trained on real data, suggesting that TABDLM can produce samples that are both distributionally consistent and beneficial for improving generalization. Among language-based models, TABDLM$_{170M}$ attains the lowest average gap. We observe a noticeable performance drop on Beijing, which may be caused by the large fraction of missing values in the target column, where our mean-imputation preprocessing could distort the true target distribution and hurt likelihood-based metrics such as MLE.

C.3 Evaluation results of α-Precision

We report α-Precision results in Table 16. TABDLM achieves the best overall score on Default and the best average score among language-based models. Compared with tabular-specific synthetic

Table 13: Per-dataset expansion of the Shape entries of Table 3. Performance comparison on the error rates (%) of **Shape** ↓. (**B**: best overall; **B**: best in language-based model)

Method	Adult	Default	Shoppers	Magic	Beijing	Average
CTGAN	16.84 ± 0.03	16.83 ± 0.04	21.15 ± 0.10	9.81 ± 0.08	21.39 ± 0.05	17.20
TVAE	14.22 ± 0.08	10.17 ± 0.05	24.51 ± 0.06	8.25 ± 0.06	19.16 ± 0.06	15.26
GOGGLE	16.97	17.02	22.33	1.90	16.93	15.03
STaSy	11.29 ± 0.06	5.77 ± 0.06	9.37 ± 0.09	6.29 ± 0.13	6.71 ± 0.03	7.89
CoDi	21.38 ± 0.06	15.77 ± 0.07	31.84 ± 0.05	11.56 ± 0.26	16.94 ± 0.02	19.50
TabDDPM	1.75 ± 0.03	1.57 ± 0.08	2.72 ± 0.13	1.01 ± 0.09	1.30 ± 0.03	1.67
TabSyn	0.81 ± 0.05	1.01 ± 0.08	1.44 ± 0.07	1.03 ± 0.14	1.26 ± 0.05	1.11
TabDiff	**0.63 ± 0.05**	1.24 ± 0.07	1.28 ± 0.09	0.78 ± 0.08	1.03 ± 0.05	0.99
TabNAT	0.73	**0.80**	**1.11**	**0.62**	**0.79**	**0.81**
GReaT	12.12 ± 0.04	19.94 ± 0.06	14.51 ± 0.12	16.16 ± 0.09	8.25 ± 0.12	14.20
DiffLM	9.74	9.06	10.07	7.53	6.35	8.55
Qwen2.5-7B$_{\text{ICL}}$	27.80	22.39	34.37	13.41	31.83	25.96
Qwen2.5-14B$_{\text{ICL}}$	25.17	21.52	51.21	17.51	29.76	29.03
Qwen2.5-7B$_{\text{SFT}}$	5.11	2.58	8.76	7.53	7.49	6.29
LLaDA-8B$_{\text{SFT}}$	1.69	2.00	13.56	5.60	3.64	5.30
TabDLM$_{\text{170M}}$	1.49	3.64	4.54	**1.96**	**2.39**	2.80
TabDLM	**1.46**	**1.18**	**2.05**	2.93	2.42	**2.01**

Table 14: Per-dataset expansion of the Trend entries of Table 3. Performance comparison on the error rates (%) of **Trend** ↓.(**B**: best overall; **B**: best in language-based model)

Method	Adult	Default	Shoppers	Magic	Beijing	Average
CTGAN	20.23 ± 1.20	26.95 ± 0.93	13.08 ± 0.16	7.00 ± 0.19	22.95 ± 0.08	18.04
TVAE	14.15 ± 0.88	19.50 ± 0.95	18.67 ± 0.38	5.82 ± 0.49	18.01 ± 0.08	15.23
GOGGLE	45.29	21.94	23.90	9.47	45.94	29.31
STaSy	14.51 ± 0.25	5.96 ± 0.26	8.49 ± 0.15	6.61 ± 0.53	8.00 ± 0.10	8.71
CoDi	22.49 ± 0.08	68.41 ± 0.05	17.78 ± 0.11	6.53 ± 0.25	7.07 ± 0.15	24.46
TabDDPM	3.01 ± 0.25	4.89 ± 0.10	6.61 ± 0.16	1.70 ± 0.22	2.71 ± 0.09	3.78
TabSyn	1.93 ± 0.07	2.81 ± 0.48	2.13 ± 0.10	0.88 ± 0.18	3.13 ± 0.34	2.18
TabDiff	**1.49 ± 0.16**	2.55 ± 0.75	1.74 ± 0.08	**0.76 ± 0.12**	2.59 ± 0.15	**1.83**
TabNAT	1.61	**2.21**	**1.67**	1.37	**2.48**	1.87
GReaT	17.59 ± 0.22	70.02 ± 0.12	45.16 ± 0.18	10.23 ± 0.40	59.60 ± 0.55	40.52
Qwen2.5-7B$_{\text{ICL}}$	37.61	31.02	34.77	20.62	40.83	32.97
Qwen2.5-14B$_{\text{ICL}}$	42.67	28.72	42.32	18.11	38.20	34.00
Qwen2.5-7B$_{\text{SFT}}$	17.03	16.99	10.28	14.45	28.68	17.49
LLaDA-8B$_{\text{SFT}}$	26.15	26.30	16.42	13.71	27.90	22.10
TabDLM$_{\text{170M}}$	**2.56**	6.19	4.62	**1.76**	**2.71**	3.57
TabDLM	2.74	**2.33**	**2.40**	2.85	2.93	**2.65**

models that are designed exclusively for numerical and categorical features, TabDLM remains competitive while uniquely supporting free-form text generation.

C.4 Evaluation results of C2ST

Table 17 reports the C2ST detection score, where higher values indicate that synthetic samples are statistically harder to distinguish from real ones. TabDLM achieves the best average result among all language-based models, improving the average C2ST score by 13.2% compared with the strongest language-based baseline. Compared with tabular-specific diffusion models designed exclusively for numerical and categorical features, TabDLM still achieves competitive performance.

The C2ST scores in Table 17 follow the standard protocol with a logistic-regression discriminator. Table 18 repeats the test with two stronger, tuned discriminators, a random forest (RF) and gradient-boosted decision trees (GBDT), keeping the column set, encoding, folds, and scoring rule unchanged. Stronger discriminators lower the score of every system, and TabDLM retains the highest average among language-model generators under all three.

Table 15: Evaluation of MLE. AUC is used for classification tasks and RMSE for regression tasks. (**B**: best overall; **B**: best in language-based model)

Methods	Adult	Default	Shoppers	Magic	Beijing	Average Gap
	AUC↑	AUC↑	AUC↑	AUC↑	RMSE↓	%
Real	.927 ± .000	.770 ± .005	.926 ± .001	.946 ± .001	.423 ± .003	0.0
CTGAN	.886 ± .002	.696 ± .005	.875 ± .009	.855 ± .006	.902 ± .019	28.48
TVAE	.878 ± .004	.724 ± .005	.871 ± .006	.887 ± .003	.770 ± .011	21.09
GOGGLE	.778 ± .012	.584 ± .005	.658 ± .052	.654 ± .024	1.09 ± .025	51.54
STaSy	.906 ± .001	.752 ± .006	.914 ± .005	.934 ± .003	.656 ± .014	12.45
CoDi	.871 ± .006	.525 ± .006	.865 ± .006	.932 ± .003	.818 ± .021	27.86
TabDDPM	.907 ± .001	.758 ± .004	.918 ± .005	.935 ± .003	.592 ± .011	9.14
TabSyn	.909 ± .001	.763 ± .002	.914 ± .004	**.937 ± .002**	.580 ± .009	8.44
TabDiff	.912 ± .002	.763 ± .005	.921 ± .004	.936 ± .003	**.555 ± .013**	**7.07**
TabNAT	.904	.764	.916	.935	.579	8.48
GReaT	.913 ± .003	.755 ± .006	.902 ± .005	.888 ± .008	.653 ± .013	13.31
DiffLM	.906	**.794**	.915	.917	.696	13.59
Qwen2.5-7B$_{\text{ICL}}$	.853	.390	.811	.829	.991	43.28
Qwen2.5-14B$_{\text{ICL}}$	.852	.639	.640	.844	1.03	42.05
Qwen2.5-7B$_{\text{SFT}}$	**.915**	.769	.895	.918	.674	13.41
LLaDA-8B$_{\text{SFT}}$	.909	.774	.872	**.919**	.679	14.13
TabDLM$_{\text{170M}}$	.908	.786	.908	.915	**.651**	**11.82**
TabDLM	.907	.791	**.923**	.905	.696	13.73

Table 16: Evaluation on α-Precision scores. (**B**: best overall; **B**: best in language-based model)

Method	Adult	Default	Shoppers	Magic	Beijing	Average
CTGAN	77.74 ± 0.15	62.08 ± 0.08	76.97 ± 0.39	86.90 ± 0.22	96.27 ± 0.14	79.99
TVAE	98.17 ± 0.17	85.57 ± 0.34	58.19 ± 0.26	86.19 ± 0.48	97.20 ± 0.10	85.06
GOGGLE	50.68	68.89	86.95	90.88	88.81	77.24
STaSy	82.87 ± 0.26	90.48 ± 0.11	89.65 ± 0.25	86.56 ± 0.19	89.16 ± 0.12	87.74
CoDi	77.58 ± 0.45	82.38 ± 0.15	94.95 ± 0.35	85.01 ± 0.36	98.13 ± 0.38	87.61
TabDDPM	96.36 ± 0.20	97.59 ± 0.36	88.55 ± 0.68	98.59 ± 0.17	97.93 ± 0.30	95.80
TabSyn	**99.39 ± 0.18**	98.65 ± 0.23	98.36 ± 0.52	99.42 ± 0.28	97.51 ± 0.24	98.67
TabDiff	99.02 ± 0.20	98.49 ± 0.28	**99.11 ± 0.34**	99.47 ± 0.21	98.06 ± 0.24	98.83
TabNAT	98.67	99.27	97.67	**99.50**	**99.27**	**98.88**
GReaT	55.79 ± 0.03	85.90 ± 0.17	78.88 ± 0.13	85.46 ± 0.54	**98.32 ± 0.22**	80.87
Qwen2.5-7B$_{\text{ICL}}$	85.51	88.56	27.96	83.39	63.76	69.84
Qwen2.5-14B$_{\text{ICL}}$	68.53	83.17	8.93	89.99	84.53	67.03
Qwen2.5-7B$_{\text{SFT}}$	84.70	95.25	95.46	82.45	94.35	90.44
LLaDA-8B$_{\text{SFT}}$	**99.32**	97.68	68.73	85.09	95.13	89.19
TabDLM$_{\text{170M}}$	96.51	88.53	97.23	95.04	97.62	94.99
TabDLM	97.62	**99.56**	**98.51**	**97.90**	97.87	**98.29**

Table 17: Evaluation on C2ST. (**B**: best overall; **B**: best in language-based model)

Method	Adult	Default	Shoppers	Magic	Beijing	Average
CTGAN	0.5949	0.4875	0.7488	0.6728	0.7531	0.6514
TVAE	0.6315	0.6547	0.2962	0.7706	0.8659	0.6438
GOGGLE	0.1114	0.5163	0.1418	0.9526	0.4779	0.4400
STaSy	0.4054	0.6814	0.5482	0.6939	0.7922	0.6242
CoDi	0.2077	0.4595	0.2784	0.7206	0.7177	0.4768
TabDDPM	0.9755	0.9712	0.8349	**0.9998**	0.9513	0.9465
TabSyn	0.9910	**0.9826**	0.9662	0.9960	0.9528	0.9777
TabDiff	**0.9950**	0.9774	**0.9843**	0.9989	0.9781	**0.9867**
TabNAT	0.9870	0.9657	0.9626	0.9989	**0.9845**	0.9797
GReaT	0.5376	0.4710	0.4285	0.4326	0.6893	0.5118
Qwen2.5-7B$_{\text{ICL}}$	0.0925	0.3089	0.0591	0.5119	0.1787	0.2302
Qwen2.5-14B$_{\text{ICL}}$	0.1564	0.4866	0.0186	0.5903	0.2702	0.3044
Qwen2.5-7B$_{\text{SFT}}$	0.8037	0.9131	0.5722	0.7845	0.7315	0.7610
LLaDA-8B$_{\text{SFT}}$	0.9366	0.9140	0.5171	0.8489	**0.9799**	0.8393
TabDLM$_{\text{170M}}$	**0.9491**	0.7790	0.8255	**0.9736**	0.9527	0.8960
TabDLM	0.9386	**0.9734**	**0.9594**	0.9254	0.9528	**0.9499**

Table 18: Evaluation on C2ST using logistic regression (LR), random forest (RF), and gradient-boosted trees (GBDT) as the classifier.

Method	Adult			Shoppers			Default			Average		
	LR	RF	GBDT	LR	RF	GBDT	LR	RF	GBDT	LR	RF	GBDT
Qwen2.5-7B$_{\text{ICL}}$	.093	.004	.001	.059	.010	.005	.309	.006	.001	.153	.007	.002
Qwen2.5-14B$_{\text{ICL}}$	.156	.013	.004	.019	.000	.000	.487	.004	.001	.221	.006	.002
Qwen2.5-7B$_{\text{SFT}}$	.804	.612	.560	.572	.314	.248	.913	**.630**	**.548**	.763	.519	.452
LLaDA-8B$_{\text{SFT}}$	.937	.737	**.753**	.517	.132	.089	.914	.564	.547	.789	.478	.463
TabDLM	**.939**	**.753**	.744	**.959**	**.591**	**.473**	**.973**	.532	.418	**.957**	**.625**	**.545**

C.5 Evaluation results of DCR

Table 19 reports the DCR score, which serves as a privacy metric: a value closer to 50% indicates that synthetic samples are equally close to training and held-out test records, suggesting the model has learned the underlying distribution rather than memorized training instances. TabDLM achieves the best overall average deviation across all baselines and obtains the best per-dataset DCR on Default and Beijing. On average, the large pretrained backbone therefore does not lead to more memorization than specialized tabular diffusion models.

The DCR analysis above covers tables without free-form text. We extend it to Amazon. Following the same protocol, the real data is split into two equal disjoint halves, one used for training and one held out, and we report the fraction of synthetic rows that are closer to the training half; values closer to 50% are better. Distances over numerical and categorical columns are computed as above; for the free-form text columns we use the cosine distance between term frequency–inverse document frequency (TF-IDF) vectors [36] computed over single words and pairs of consecutive words. As shown in Table 20, TabDLM is the closest to 50% on the structured channel and on average, whereas Qwen2.5-7B$_{\text{SFT}}$ is saturated on both channels, i.e., it largely reproduces its training set. Table 21 additionally measures verbatim reuse, averaged over the three longest free-form text columns (`description`, `review_text` and `summary`): the exact-duplicate rate, the fraction of generations whose TF-IDF similarity to a training text exceeds 0.8, and the fraction of generated n-grams (runs of n consecutive words) that occur verbatim in the training text. A held-out real split, scored identically, gives the level of reuse that is natural for this corpus. TabDLM has the lowest long-span reuse among all fine-tuned systems.

Table 19: Evaluation on DCR score, where a score closer to 50% is more preferable. We also report the average absolute deviation of the DCR score from 50% across all datasets, where lower values are better. (**B**: best overall)

Method	Adult	Default	Shoppers	Beijing	Average ↓
STaSy	$50.33\% \pm 0.19$	$50.23\% \pm 0.09$	$51.53\% \pm 0.16$	$50.59\% \pm 0.29$	0.67%
CoDi	**$49.92\% \pm 0.18$**	$51.82\% \pm 0.26$	$51.06\% \pm 0.18$	$50.87\% \pm 0.11$	0.96%
TabDDPM	$51.14\% \pm 0.18$	$52.15\% \pm 0.20$	$63.23\% \pm 0.25$	$80.11\% \pm 2.68$	11.66%
TabSyn	$50.94\% \pm 0.17$	$51.20\% \pm 0.18$	$52.90\% \pm 0.22$	$50.37\% \pm 0.13$	1.35%
TabDiff	$50.10\% \pm 0.32$	$51.11\% \pm 0.36$	**$50.24\% \pm 0.62$**	$50.50\% \pm 0.36$	0.49%
TabDLM	50.73%	**49.95%**	50.87%	**50.13%**	**0.45%**

Table 20: DCR on Amazon, separately for the structured columns and the free-form text columns. A score closer to 50% is preferable.

Method	Num.+Cat.	Text	Mean $\lvert\Delta$ from $50\%\rvert$ ↓
Qwen2.5-7B$_{\text{ICL}}$	41.04%	37.93%	10.52%
Qwen2.5-14B$_{\text{ICL}}$	44.68%	**42.01%**	6.66%
Qwen2.5-7B$_{\text{SFT}}$	98.16%	99.85%	49.01%
LLaDA-8B$_{\text{SFT}}$	48.05%	61.36%	6.66%
TabDLM	**50.24%**	59.37%	**4.81%**

Table 21: Verbatim reuse of training text on Amazon (lower is less reuse), averaged over the free-form text columns.

Method	Exact dup. ↓	Sim. > 0.8 ↓	5-gram ↓	8-gram ↓	13-gram ↓
Real held-out split (reference)	29.78%	39.98%	13.96%	12.20%	11.51%
Qwen2.5-7B$_{\text{ICL}}$	23.28%	**8.00%**	**0.55%**	0.21%	0.15%
Qwen2.5-14B$_{\text{ICL}}$	19.79%	8.99%	1.98%	0.22%	0.04%
Qwen2.5-7B$_{\text{SFT}}$	36.26%	38.67%	9.76%	2.97%	2.07%
LLaDA-8B$_{\text{SFT}}$	28.56%	33.96%	2.15%	0.18%	0.02%
TabDLM	**18.27%**	24.04%	1.03%	**0.07%**	**0.00%**

C.6 Distributions of the numerical columns

Figures 3 and 4 compare the distribution of every numerical column of Amazon and Arxiv in the generated rows with that in the real data. On the long-tailed `price` of Amazon, DiffLM and the Qwen2.5 baselines cut the tail: none of them generates a `price` above 66, versus 1,000 in the real data, whereas TabDLM covers the full range of the real prices. On the other three columns most systems, including GReaT, are close to the real data, but some language-model baselines generate values far beyond the real maximum: 58% of the `abstract_length` values generated by Qwen2.5-14B$_{\text{ICL}}$ exceed the real maximum of 109, and LLaDA-8B$_{\text{SFT}}$, which shares the backbone of TabDLM but writes numbers as digit tokens, generates `abstract_length` values of up to 9,013. Neither variant of TabDLM generates a value outside the real range on any of the four columns.

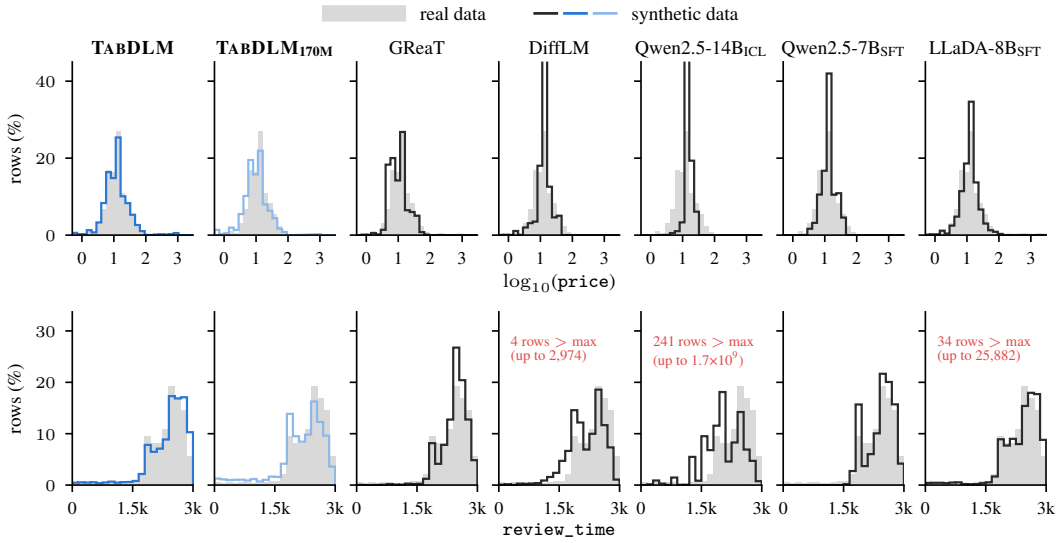


Figure 3: Distributions of the numerical columns on Amazon (real maximum: `price` 1,000, `review_time` 2,922). Red notes count synthetic values above the real maximum; bars taller than the axis are cut at the top.

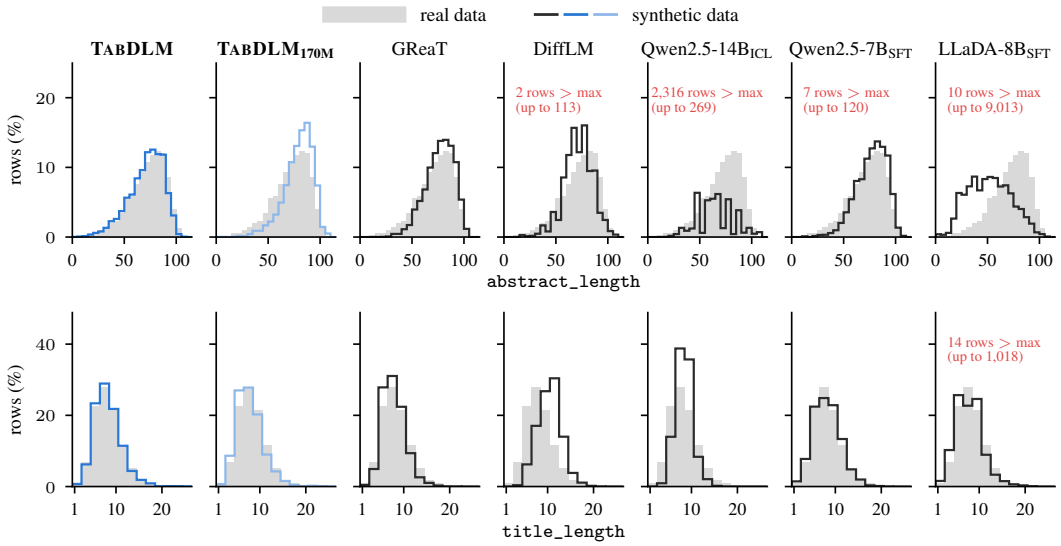


Figure 4: Distributions of the numerical columns on Arxiv (real maximum: `abstract_length` 109, `title_length` 27), as in Figure 3.

C.7 Quality of the generated free-form text

We evaluate the free-form text of the generated Amazon rows with four metrics. Distinct-2 (the fraction of unique bigrams) and Self-BLEU measure the diversity of the generated text. An LLM judge (`gpt-5-mini`) rates each row on a 1–5 scale for cross-field consistency and lists every statement in a text field that contradicts a numerical or categorical column of the same row; the hallucination rate is the fraction of rows with at least one such contradiction. Rows from all systems and real rows are pooled, shuffled and judged blind, five rows per request. As shown in Figure 5, TabDLM attains the highest Distinct-2 and judged consistency and the lowest Self-BLEU (tied with TabDLM$_{170M}$).

	Distinct-2 ↑	Self-BLEU ↓	Consistency ↑	Hallucination% ↓
Real data	0.546	0.016	4.79	8.3
GReaT	0.320	0.097	4.57	18.7
DiffLM	0.286	0.073	4.78	8.3
Qwen2.5-7B$_{ICL}$	0.262	0.075	4.80	9.7
Qwen2.5-14B$_{ICL}$	0.307	0.052	4.85	**4.0**
Qwen2.5-7B$_{SFT}$	0.328	0.036	4.03	17.3
LLaDA-8B$_{SFT}$	0.518	0.002	4.53	12.0
TabDLM$_{170M}$	0.620	**0.001**	3.74	17.7
TabDLM	**0.673**	**0.001**	**4.86**	7.0

better
worse

Figure 5: Quality of the generated free-form text on Amazon. Consistency is the LLM-judge score (1–5); Hallucination% is the percentage of rows whose text contradicts a numerical or categorical column. Within each column, bluer is better and the best value is in bold.

The judge prompt is given below: the system message, then the user message with the rubric and the rows to judge.

LLM-Judge Prompt

System

You are a careful data-quality annotator for a table of Amazon product reviews.

Your job is to judge whether the fields of each row are mutually consistent and whether the free-text fields are grounded in the structured fields. You are NOT judging whether the review is well-written, helpful, or a product you would buy.

Critically: real Amazon reviews are often terse, ungrammatical, lowercase, off-topic, or oddly punctuated, and real ratings are sometimes a poor match for the review's tone. None of that is a defect here. Penalize only genuine cross-field contradictions and genuine incoherence – never mere brevity, informality, or an unremarkable product.

Judge only what is present in the row. Do not use outside knowledge about whether a product or brand really exists.

Return only JSON matching the provided schema.

User

Each row is one product review. Fields:

`price`	numeric product price
`rating`	integer 1–5, the reviewer's star rating
`category`	product category
`verified`	whether the purchase was verified
`brand`, `title`	the product's brand and product title
`description`	the product's description text
`review_text`	the reviewer's free-text review
`summary`	the reviewer's short headline for the review

Score each dimension 1–5 (integers only).

1) `rating_consistency` – does the sentiment of `review_text`/`summary` match `rating`?

5 clearly matches (5 stars with praise; 1 star with complaints; 3 stars mixed)
4 matches, with minor ambiguity
3 unclear or genuinely mixed; no outright contradiction
2 probable mismatch (rating 5, review mostly negative)
1 flat contradiction (rating 5 with "broke immediately, do not buy")

If `review_text` is empty or contentless, score 3 and say so – do not guess.

2) `product_consistency` – do brand / title / description / category describe ONE coherent product, and does `review_text` review THAT product?

5 all fields point to the same product
4 coherent; one field vague or generic
3 some drift, still plausibly the same product
2 a field clearly describes a different product type
1 fields are unrelated (title a kitchen tool, review about a paperback)

3) `summary_entailment` – is `summary` supported by `review_text`?

5 summary is a faithful compression of the review
4 faithful, slightly broader or narrower
3 related but adds or omits something notable
2 summary asserts something the review does not support
1 summary contradicts the review, or describes a different experience

If either field is empty, score 3.

4) `intra_text_coherence` – are description / review_text / summary each internally coherent as writing?

5 reads as a real human-written passage
4 minor awkwardness
3 noticeably odd but followable
2 disjointed, self-contradicting, or looping
1 degenerate: repeated n-grams, truncated mid-word, word salad

Terse-but-sensible text is a 5, not a 2. Length is not a defect.

5) `hallucinations` – list every SPECIFIC assertion in the text fields that CONTRADICTS a structured field of this same row. For each, quote the span verbatim and name the column it contradicts.

Examples that count: the review says "for the price of under $5" while price is 89.99; the review says "this Nikon lens" while brand is Logitech; the review says "arrived in the Kindle store" while category is Pet Supplies. Does NOT count: any detail that is merely unverifiable (an invented model number, a made-up anecdote, a fictional-sounding brand). Unverifiable is not contradiction – this list must stay checkable, so contradiction with a column in this row is the only criterion. Empty list is the common case.

Rows to judge (each a JSON object). They come from different sources in no particular order; judge each on its own.

{one JSON object per row: `row_id`, `price`, `rating`, `category`, `verified`, `brand`, `title`, `description`, `review_text`, `summary`}

Return JSON with key `results`: exactly 5 elements, one per `row_id` above.

Assistant

C.8 Case study

The boxes below show randomly sampled rows generated by each system on Arxiv. In the real data, `title_length` and `abstract_length` are exactly the numbers of words in `title` and `abstract`, and `category` is the primary arXiv category of the paper, so all three can be checked against the text of the same row. Values that contradict the text of their row or lie outside the range of the real data, as well as repeated or malformed passages of the text, are marked in red and explained below the row; orange marks a minor deviation.

TabDLM

submission_time	submission_year	title_length	abstract_length	category	arxiv_code
891	2020	5	77	gr-qc	arXiv:2006.09404

`title`: Quantum Charges around Static Spheres

`abstract`: We introduce a new effective formalism to calculate the potential seen by quantum fields around a static object. First proposed by Schwinger, this approach refers to a Minkowski reference space and provides a ready-to-use calculation of the potential using any quantum field theory. We show that within the framework of asymptotic safety, one can deduce the gravitational potential up to second order in the metric perturbation from this potential. A numerical illustration of our method is provided.

✓ `title_length` and `abstract_length` match the title (5 words) and the abstract (77 words), and the paper fits its `category`.

submission_time	submission_year	title_length	abstract_length	category	arxiv_code
1405	2021	6	71	quant-ph	arXiv:2101.00211

`title`: Causal effects of interacting quantum systems

`abstract`: We consider mutually entangled quantum subsystems whose total state is repeatedly prepared and measured, and they are taken to be dynamically relaxing down to an asymptotic steady state where measurements yield all coherent contributions. We show that the dynamics of at most one subsystem makes the dynamics dependent on the trajectory of another subsystem, and this dependence persists when conditioned on measurement. These results demonstrate "causal effects" of interacting quantum systems.

✓ `title_length` and `abstract_length` match the title (6 words) and the abstract (71 words), and the paper fits its `category`.

TabDLM$_{170M}$

submission_time	submission_year	title_length	abstract_length	category	arxiv_code
973	2020	4	89	hep-th	arXiv:2009.06098

`title`: Effective D variable analysis

`abstract`: We study the static D field equations in the cosmological Nambu–Walker spacetime with arbitrary proper time, having consequences which are not unitary local. When the boundary of D fields is changed, a directly interacting Newtonian term emerges appearing in the resulting action that can be removed, by a remarkable operator dependent hypergeometric transformation. As fixed points is expanded, proper-time CPT disappears, which leads to vanishing of the additional term. The resulting action preserves all conformal dependence and hence provide a direct, unitary, solution with no trace of nonlocal boundary conditions.

✓ `title_length` matches the title (4 words), and the paper fits its `category`.
▲ `abstract_length` is 89, one word fewer than the abstract (90 words).

GReaT

submission_time	submission_year	title_length	abstract_length	category	arxiv_code
1625	2022	8	95	hep-ph	arXiv:2207.08231

`title`: Black Hole Thermodynamics in D=2

`abstract`: We consider the thermodynamic properties of a two-dimensional black hole in the framework of the D=2 conformal field theory. In the first solution, the thermodynamic entropy is obtained by the Ramsey scheme, while in the second solution it depends on the thermodynamic entropy. Under the Ramsey scheme, the entropy becomes a dependent quantity in the thermodynamic field theory. Under the inverted symmetries, the thermodynamic entropy, under the inverted symmetries, depends on the parameters of the inverted symmetries. Under the inverted symmetries, the thermodynamic entropy depends on the parameters of the inverted symmetries.

✗ `title_length` is 8, but the title has 5 words.
✗ The last two sentences say the same thing and repeat "the inverted symmetries" five times.

DiffLM

submission_time	submission_year	title_length	abstract_length	category	arxiv_code
1655	2022	13	60	quant-ph	arXiv:2206.11744

title: A new type of quantum error correction for quantum networks

abstract: Quantum error correction is a fundamental tool for error-free quantum communication in a quantum network. We propose a new type of quantum error correction scheme, which is based on the quantum teleportation protocol, and can be used to correct errors in quantum networks. The scheme is simple, practical, and can be easily extended to correct errors in quantum networks with more than two nodes. The proposed scheme can be used to correct errors in a quantum network with only two nodes, and it can also be used to correct errors in a quantum network with more than two nodes.

✗ `title_length` is 13 and `abstract_length` is 60, but the title has 10 words and the abstract 99.
✗ The last sentence repeats the previous one.

Qwen2.5-7B$_{\text{ICL}}$

submission_time	submission_year	title_length	abstract_length	category	arxiv_code
365	2023	12	150	astro-ph.SR	arXiv:2303.12345

title: Exploring Quantum Entanglement in Black Hole Horizons

abstract: This paper delves into the quantum entanglement properties near black hole horizons, presenting new theoretical models that predict enhanced entanglement due to Hawking radiation. The study uses advanced computational techniques to simulate these effects and discusses potential implications for quantum gravity theories.

✗ `title_length` is 12 and `abstract_length` is 150, but the title has 7 words and the abstract 42; 150 also exceeds the longest real abstract (109 words).
✗ `submission_year` is 2023, outside the years of the real data (2018–2022).
✗ `category` is `astro-ph.SR` (solar and stellar astrophysics), but the paper is about quantum entanglement at black-hole horizons.
✗ The same title appears in 241 and the same `arxiv_code` in 597 of the 4,000 generated rows.

Qwen2.5-14B$_{\text{ICL}}$

submission_time	submission_year	title_length	abstract_length	category	arxiv_code
365	2022	9	120	cond-mat.mes-hall	arXiv:2208.01234

title: Exploring Quantum Entanglement in Graphene Bilayers

abstract: Recent theoretical studies predict that quantum entanglement can significantly enhance the electronic properties of graphene bilayers. This paper explores these predictions through advanced computational models, providing insights into the potential applications in quantum computing and nanotechnology.

✗ `title_length` is 9 and `abstract_length` is 120, but the title has 6 words and the abstract 36; 120 also exceeds the longest real abstract (109 words).
✗ The same title appears in 200 of the 4,000 generated rows, and 767 generated `arxiv_code` values end in `01234`.

Qwen2.5-7B$_{\text{SFT}}$

submission_time	submission_year	title_length	abstract_length	category	arxiv_code
1117	2021	6	90	hep-th	arXiv:2004.07125

title: Perturbative effects of a magnetic field on the dispersion relations of Bloch electrons

abstract: We study the effect of a magnetic field on the dispersion relations of electrons in a two-dimensional lattice under the influence of a generalized Peierls–Aharonov and Berry phases. The latter is constructed from the Berry connection of Bloch electrons in the high symmetry points of the Brillouin zone. We analyze the perturbative effects of magnetic field on the dispersion relations of electron states and find the conditions for the source of the topological states at the zone boundary.

✗ `title_length` is 6 and `abstract_length` is 90, but the title has 13 words and the abstract 78.
✗ `category` is `hep-th` (high-energy physics theory), but the paper is about electrons in a crystal lattice, a condensed-matter topic.

LLaDA-8B$_{\text{SFT}}$

submission_time	submission_year	title_length	abstract_length	category	arxiv_code
35	2018	9	50	quant-ph	arXiv:1802.01802

title: Coordinate Independence and Dimension Dimensionality in FriedmannEquations

abstract: In the present work, we explore the relation between the generalized dimensions and generalized anisotropic space model in the Friedmann Equation. The solution of the generalized Friedmann equation and the generalized and the anisotropic Friedmann equations are obtained. We also find the anisotropic and generalized static solutions for the static case for both the generalized and anisotropic Friedmann equations.

✗ `title_length` is 9 and `abstract_length` is 50, but the title has 7 words and the abstract 59.
✗ Two words of the title are run together.
✗ `category` is `quant-ph` (quantum physics), but the paper is about the Friedmann equations of cosmology.

In both rows of TABDLM, `title_length` and `abstract_length` match the text exactly, the paper fits its `category`, and the abstract is coherent and free of repetition. TABDLM$_{170M}$ makes one minor slip, an `abstract_length` one word shorter than its abstract, which we attribute to its much smaller backbone. Every baseline row, in contrast, contains at least one clear error. The most common is a length value that contradicts the text of its own row: Qwen2.5-7B$_{ICL}$ reports a 150-word abstract for an abstract of 42 words, and LLaDA-8B$_{SFT}$, which shares the backbone of TABDLM but writes numbers as digit tokens, misreports both lengths, in line with Figure 2. The text of the baselines degrades as well: GReaT and DiffLM repeat the same sentence, and the in-context baselines fall back on a few templates, reusing the same title in hundreds of rows.

C.9 Description of Ablation Variants

The variants in the ablation study are defined as follows. (i) TABDLM-noFloatAE replaces the pre-trained and frozen float encoder/decoder with randomly initialized modules of the same architecture, trained jointly with the rest of the model. (ii) TABDLM-onlyContDiff converts categorical features into one-hot vectors and feeds them through the continuous diffusion branch, removing MDLM-based modeling for categorical fields. (iii) TABDLM-onlyMDLM serializes numerical features and processes them in the same way as categorical and textual features, removing the continuous diffusion branch entirely. The aggregated results are reported in Table 4.

D Limitations and Future Directions

First, the sampling efficiency of TABDLM is lower than that of existing tabular data generation methods such as TabDiff [38] and TabSyn [49]. This limitation can be primarily attributed to the large model size of the MDLM backbone. Nevertheless, TABDLM is designed for broader application scenarios, as it supports the generation of numerical, categorical, and free-form text fields, which specialized tabular diffusion models do not support. Moreover, common methods for accelerating MDLM inference, like KV caching [44], could be incorporated to improve sampling efficiency. However, it is out of the scope of this work. Second, TABDLM models numerical and language diffusion using a shared noise schedule that couples the denoising dynamics across modalities and enforces a common step size for denoising both numerical and textual features. While effective in practice, this design may be suboptimal in certain scenarios. Introducing modality-specific noise schedules to partially decouple the denoising processes can be promising, which we leave for future investigation.

E Societal Impact

TABDLM enables high-fidelity synthetic tabular data generation across numerical, categorical, and free-form text fields, which can support privacy-preserving data sharing, data augmentation in low-resource settings, and controlled benchmarks for tabular foundation model research. The DCR results in Appendix C.5 suggest that TABDLM learns the underlying distribution rather than memorizing training records, which is desirable for these use cases. As with any generative model, care should be taken when applying TABDLM beyond the evaluated benchmarks. Synthetic rows that jointly couple structured attributes and fluent text could be misused for fabricated reviews or profiles, and generated data may inherit biases from the training distribution. High-stakes deployments should involve human oversight, privacy auditing, and bias evaluation on both the source data and the generated samples.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The claims in the abstract and introduction accurately reflect our contributions and scope.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Limitations are discussed in Appendix D.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

Justification: This paper does not include theoretical results.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The model architecture, training, and sampling procedures are described in Section 2, with full implementation details and hyperparameters in Appendix B and dataset construction in Appendix A.1.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: Code is released at the URL provided in Appendix B. Synthetic dataset construction is fully specified in Appendix A.1; all real-world datasets are publicly available.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

Answer: [Yes]

Justification: Training and evaluation details are provided in Appendix A and Appendix B.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: We report mean $\pm$ standard deviation over multiple runs for baselines reproduced from prior work (Appendix C.1). For language-model-based methods, we use a single fixed random seed shared across all methods due to compute cost, as documented in Appendix B.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: All experiments are run on a single NVIDIA A100 GPU (80GB), with per-dataset training epochs reported in Appendix B.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conforms to the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Both positive and negative societal impacts, along with mitigations, are discussed in Appendix E.

    Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [Yes]

    Justification: We evaluate memorization risk via the DCR metric (Appendix C.5), and ProfileBio is constructed without sensitive personal attributes (Appendix A.1.2). Further safeguards are discussed in Appendix E.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All datasets and pretrained models are cited to their original sources and used in accordance with their public licenses.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.

- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The two new synthetic datasets, MathExpr and ProfileBio, are documented in Appendix A.1.1 and A.1.2. Source code is released at the URL in Appendix B.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: This work does not involve crowdsourcing or human subjects research.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: This work does not involve human subjects research.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: TABDLM is built on a pretrained masked diffusion language model (LLaDA-8B), as described in Section 2 and Appendix B. The text-quality evaluation in Appendix C.7 uses an LLM judge (`gpt-5-mini`), whose full prompt is given there.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.