CDMD: A CROSS-DATASET MIXED-TYPE DIFFUSION MODEL FOR TABULAR DATA

Anonymous authors
Paper under double-blind review

ABSTRACT

Generative models for tabular data are typically trained separately for each dataset, limiting knowledge transfer and requiring the storage of many specialized models. In this paper, we introduce CDMD, a tabular diffusion model trained jointly across heterogeneous datasets with different schemas and variable numbers of numerical and categorical features. Unlike existing cross-dataset tabular diffusion models that operate in continuous representation spaces, CDMD defines diffusion directly over the mixed-type feature space and is trained end-to-end. To accommodate heterogeneous categorical domains, we introduce a schema-restricted reverse-process parameterization for masked diffusion models, in which the output space dynamically adapts to each feature's vocabulary. We then compose numerical and categorical feature-level diffusion processes into a schema-dependent row-level process. A shared schema-aware Transformer denoiser captures dependencies between features and parameterizes the reverse process across varying schemas. On seven real-world datasets, a single jointly trained CDMD achieves the highest average generation quality among strong single-dataset and cross-dataset baselines, while using substantially fewer total parameters than the collection of separately trained models. Furthermore, pre-training on a corpus of 337 datasets improves generation on previously unseen datasets under both limited target data and limited adaptation epochs. These results demonstrate the potential of direct mixed-type diffusion for shared and transferable tabular data generation.

1 INTRODUCTION

Synthetic tabular data generation has the potential to support a wide range of applications, from data augmentation and privacy-preserving data sharing to software testing and simulation (Van Breugel & Van der Schaar, 2023). Such applications are particularly relevant in healthcare, finance, and enterprise settings, where real data is often sensitive and publicly available datasets are scarce (Borisov et al., 2022a). Recent advances in tabular generative modeling, particularly with diffusion-based models, have improved the statistical fidelity of synthetic data as well as its utility for training downstream predictive models (Shi et al., 2025b; Stoian et al., 2025). However, many established tabular generative models are designed for individual datasets, requiring training and storing a separate model for each target dataset. This approach limits opportunities for cross-dataset knowledge transfer and may compromise the quality of generation on small datasets. *Cross-dataset tabular generative models* offer a promising alternative: a single model trained across multiple datasets can reduce per-dataset training and storage costs while learning reusable patterns that can support generation in limited-data settings and accelerate adaptation to new datasets (Van Breugel & Van Der Schaar, 2024). Developing such models, however, is challenging because tabular datasets vary widely in feature semantics, dimensionality, feature types, and distributions.

To address this heterogeneity, existing cross-dataset tabular generative models often represent mixed-type data in continuous spaces and learn shared diffusion models over these representations. For instance, Van Breugel et al. (2024) apply Gaussian diffusion to numerical features and to continuous high-dimensional embeddings of categorical features derived from a pre-trained language model, while Lin et al. (2025) and Han et al. (2026) encode rows from different tables into shared latent representations, perform Gaussian diffusion in the latent space, and decode generated representations back to the feature space. We discuss these approaches in more detail in Appendix A, where we also discuss methods that adapt predictive tabular foundation models for synthetic data

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

generation. Although these methods demonstrate the potential of cross-dataset generative modeling, latent-space approaches introduce additional encoding and decoding stages, which typically require two-stage training and sampling procedures and make generation quality dependent on both latent modeling and reconstruction accuracy (Lin et al., 2025). Meanwhile, Shi et al. (2025a) have demonstrated the effectiveness of mixed-type diffusion for tabular data generation, albeit in the single-dataset setting. These considerations motivate investigating cross-dataset generative modeling directly in the original mixed-type feature space.

In this work, we introduce a cross-dataset tabular diffusion model that combines continuous and discrete diffusion to model heterogeneous datasets directly in their mixed-type feature spaces. Our main technical contributions are threefold. First, we propose a simple reverse-process parametrization for masked diffusion models (Sahoo et al., 2024) that dynamically adapts to each categorical feature's vocabulary, enabling a shared diffusion formulation across heterogeneous categorical features. Second, we develop a mixed-type diffusion process that composes feature-level processes according to a dataset's schema, accommodating rows with varying numbers and types of features. Third, based on the resulting diffusion objective, we design a Transformer-based denoising network that maps noisy variable-sized mixed-type rows to corresponding predictions that parametrize the reverse denoising process. We call our model CDMD (Cross-Dataset Mixed-type Diffusion model).

We evaluate CDMD in two main settings: in-domain generation and adaptation to unseen datasets. For in-domain generation, we train CDMD jointly on seven widely used real-world datasets spanning diverse domains. A single CDMD model achieves higher average synthetic data quality than state-of-the-art single-dataset baselines, while using a fraction of their combined parameter count. For adaptation to unseen datasets, we pre-train CDMD on a corpus of 337 datasets, and evaluate adaptation with either a limited number of training samples or training epochs. Under both adaptation constraints, CDMD achieves higher generation quality than the baselines.

2 Problem formulation

Notation. We denote a corpus of L tabular datasets as $\{\mathcal{T}^{(\ell)}\}_{\ell=1}^{L}$, where each dataset $\mathcal{T}^{(\ell)} = \left(\mathcal{S}^{(\ell)}, \boldsymbol{X}^{(\ell)}\right)$ consists of a table schema $\mathcal{S}^{(\ell)}$ and a set of rows $\boldsymbol{X}^{(\ell)} = \{\boldsymbol{x}_i^{(\ell)}\}_{i=1}^{N^{(\ell)}}$. Following Van Breugel et al. (2024), we define the schema as $\mathcal{S}^{(\ell)} = \left(C^{(\ell)}, \{s_j^{(\ell)}\}_{j=1}^{d^{(\ell)}}\right)$, where $C^{(\ell)}$ is the dataset's name or description, $d^{(\ell)}$ is the number of columns (features), and $s_j^{(\ell)}$ is a column schema defining its name, type (numerical or categorical) and domain as follows:

$$s_j^{(\ell)} = \big(\underbrace{c_j^{(\ell)}}_{\text{name}}, \underbrace{\tau_j^{(\ell)}}_{\text{type}}, \underbrace{\mathcal{X}_j^{(\ell)}}_{\text{domain}}\big), \text{ where } \tau_j^{(\ell)} \in \{\text{num}, \text{cat}\} \text{ and } \mathcal{X}_j^{(\ell)} = \begin{cases} \mathbb{R} & \text{if } \tau_j^{(\ell)} = \text{num}, \\ \mathcal{V}_j^{(\ell)} & \text{if } \tau_j^{(\ell)} = \text{cat}, \end{cases}$$

with $\mathcal{V}_j^{(\ell)} = \{v_{j,k}^{(\ell)}\}_{k=1}^{K_j^{(\ell)}}$ denoting the vocabulary of categorical column j with $K_j^{(\ell)}$ distinct possible values. Following common practice in tabular data generation (Kotelnikov et al., 2023), we assume that each numerical feature is normalized using a transformation fitted only to the corresponding column of the training data. We use $\mathcal{J}_{\text{num}}^{(\ell)}$ and $\mathcal{J}_{\text{cat}}^{(\ell)}$ to refer to the index sets of numerical and categorical columns, respectively, and $\boldsymbol{x}^{\text{num}}$ and $\boldsymbol{x}^{\text{cat}}$ to refer to the numerical and categorical parts, respectively, of a row $\boldsymbol{x}$. We additionally define the row space induced by schema $\mathcal{S}^{(\ell)}$ as $\mathcal{X}_{\mathcal{S}^{(\ell)}} = \prod_{j=1}^{d^{(\ell)}} \mathcal{X}_j^{(\ell)}$. Throughout the paper, we will drop any index that is irrelevant or clear from context.

Tabular generative modeling. In the context of generative modeling for tabular data, we assume the rows of a given dataset $\mathcal{T}^{(\ell)}$ to be independent and identically distributed (iid) samples from an unknown data distribution $p_{\text{data}}^{(\ell)}$, i.e., $\{\boldsymbol{x}_i^{(\ell)}\}_{i=1}^{N^{(\ell)}} \overset{\text{iid}}{\sim} p_{\text{data}}^{(\ell)}$, and we aim to approximate it with a parametric distribution $p_{\boldsymbol{\theta}} \approx p_{\text{data}}^{(\ell)}$ that enables sampling synthetic rows. Given L datasets, *single-dataset generative models* learn a separate parametric distribution for each dataset, $p_{\boldsymbol{\theta}^{(\ell)}} \approx p_{\text{data}}^{(\ell)}$ (Xu et al., 2019; Kotelnikov et al., 2023; Shi et al., 2025a). Consequently, they often hard-code the schema information in the model architecture and parameters. For instance, they learn column-specific embedding and output layers, and the network width is controlled by the number of columns. This rigidity requires training a new model, or parts of it, from scratch on each dataset.

In contrast, *cross-dataset generative models* learn a single parametric distribution shared between datasets. One approach is by conditioning the distribution of a tabular dataset on its schema, $p_{\boldsymbol{\theta}}\left(\cdot \mid \mathcal{S}^{(\ell)}\right) \approx p_{\text{data}}^{(\ell)}$. Formally, $p_{\boldsymbol{\theta}}$ defines a family of probability distributions, mapping a schema $\mathcal{S}$ to a probability distribution over its row space $\mathcal{X}_{\mathcal{S}}$. Another approach to cross-dataset modeling is to provide sample rows from the target dataset as context to the model, as done by Han et al. (2026). A key difference between these two approaches is that in-context generators require target rows during sampling, whereas schema-conditional models can sample from a target dataset given only its schema. We focus on schema-conditional approaches in this work. To the best of our knowledge, existing cross-dataset tabular diffusion models (both in-context and schema-conditional) define their learned distributions over a continuous space shared between features and datasets (van Breugel et al., 2024; Lin et al., 2025; Han et al., 2026). In this work, we propose a cross-dataset schema-conditional *mixed-type* diffusion model that leverages both continuous and discrete diffusion.

3 Cross-Dataset Mixed-type Diffusion model (CDMD)

Overview. To address the heterogeneity across datasets without relying on a shared continuous representation, we propose a fine-to-coarse approach that dynamically combines feature-level diffusion processes into a row-level process based on the target schema. Our approach can be summarized as follows: **(1)** For each data type, numerical or categorical, we define an atomic one-dimensional diffusion process that can model an arbitrary tabular feature of that type (Section 3.1). We devote special care to addressing the heterogeneity across discrete vocabularies of categorical features. **(2)** Given a table schema $\mathcal{S}$, we compose these one-dimensional diffusion processes to *dynamically* define a d-dimensional diffusion process to model tabular rows (Section 3.2). **(3)** We parametrize the row-level reverse diffusion process using a schema-conditional, transformer-based denoising architecture (Section 3.3, Figure 3).

3.1 Feature-level diffusion processes

The first step towards our goal is to develop feature-level diffusion processes that can model any numerical or categorical feature and allow us to train a shared schema-conditional neural network on heterogeneous columns. In this section, x denotes a one-dimensional random variable representing one tabular feature with column schema s. The main idea behind diffusion models is to define a noising process that progressively corrupts the observed data and to learn the reverse denoising process, thereby enabling the sampling of new data points from pure noise (Sohl-Dickstein et al., 2015). Let $t \in [0,1]$ be the diffusion time and $\mathrm{x}_t \sim p_t$ the distribution of x at time t. Specifically, $\mathrm{x}_0 \sim p_0(\cdot \mid s)$ is distributed according to the data distribution, and $\mathrm{x}_1 \sim p_1$ is sampled from the prior distribution. We use q to refer to the diffusion kernel controlling the state transitions between two arbitrary time steps $t, t' \in [0,1]$ such that $p_t(x_t) = \mathbb{E}_{\mathrm{x}_{t'} \sim p_{t'}}[q(x_t \mid x_{t'})]$.

Gaussian diffusion for normalized numerical features. Numerical features share the same domain, $\mathbb{R}$; therefore, we can model their distributions with a shared noising process and a schema-conditional reverse process. Following prior work on tabular data generation (Zhang et al., 2024; Shi et al., 2025a), we adopt the Variance Exploding formulation of score-based generative models (Song et al., 2020) whose noising process corresponds to the Gaussian diffusion kernel

$$q_{\text{num}}\left(x_t \mid x_0\right) = \mathcal{N}\left(x_t; x_0, \sigma_t^2\right), \tag{1}$$

where $\sigma_t \colon [0,1] \to \mathbb{R}_+$ is the Gaussian diffusion schedule. To model the reverse process, we parametrize the score function of p_t as $\frac{\partial}{\partial x_t} \log p_t(x_t) \approx \frac{\mu_{\boldsymbol{\theta}}(x_t,t,s) - x_t}{\sigma_t^2}$ (Karras et al., 2022), where $\mu_{\boldsymbol{\theta}}(\cdot,\cdot,s) \colon \mathbb{R} \times [0,1] \to \mathbb{R}$ is a schema-conditional denoiser network that predicts clean data from noisy data. To train $\mu_{\boldsymbol{\theta}}$ on multiple features, we minimize the expected L_2 denoising loss,

$$\mathcal{L}_{\text{num}}(\boldsymbol{\theta}) = \mathbb{E}_{\mathrm{s} \sim p_s} \mathbb{E}_{\mathrm{x}_0 \sim p_0(\cdot \mid s)} \mathbb{E}_{\mathrm{t} \sim \mathcal{U}([0,1])} \mathbb{E}_{\mathrm{x}_t \sim q_{\text{num}}(\cdot \mid x_0)} \left[\lambda_t \left(\mu_{\boldsymbol{\theta}}(x_t,t,s) - x_0\right)^2\right], \tag{2}$$

where p_s is a distribution over the features available for training and λ_t is a time-dependent weighting function (Karras et al., 2022). Once trained, we can sample from the learned distribution corresponding to any target schema s by numerically solving the (schema-conditioned) flow ordinary differential equation (ODE) (Song et al., 2020), with σ_t' denoting the time derivative of σ_t,

$$dx = -\sigma_t' / \sigma_t \left(\mu_{\boldsymbol{\theta}}(x,t,s) - x\right) dt. \tag{3}$$

Generalized masked diffusion for categorical features. In contrast to numerical features, which share the same domain, $\mathbb{R}$, and to discrete data from other domains, such as language modeling, where tokens share a common vocabulary, tabular categorical features usually have distinct (possibly overlapping) domains. To address this heterogeneity, we develop a generalized discrete diffusion process defined over the joint space of all vocabularies of interest, which can be efficiently applied to an arbitrary number of categorical features thanks to a novel reverse-process parametrization.

In this work, we adapt masked diffusion models (Austin et al., 2021; Sahoo et al., 2024) to our setting for their simplicity and strong empirical performance in tabular data generation (Shi et al., 2025a). Formally, our generalized process is defined over $\mathcal{U}$, the universe of all categories from our datasets of interest, augmented with a special [MASK] category denoted by m, and we demonstrate how it can be applied to efficiently model an arbitrary categorical random variable x with schema s and domain $\mathcal{V} \subset \mathcal{U}$. In masked diffusion models, the noising process is an absorbing process that sets the clean value x_0 to the masked state m with probability $1-\alpha_t$ by time t, where $\alpha_t \colon [0,1] \to [0,1]$ is the masked diffusion schedule. The masked diffusion kernel can be written as (Sahoo et al., 2024)

$$q_{\text{cat}}(x_t \mid x_0) = \alpha_t \mathbf{1}_{x_t = x_0} + (1-\alpha_t)\mathbf{1}_{x_t = m}, \tag{4}$$

where $\mathbf{1}_{\text{condition}}$ is the indicator function, evaluating to 1 if condition is true, and 0 otherwise. This formulation implies that for all $t \in [0,1]$, x_t can either be m or x_0. The true posterior distribution from t to $t' < t$, conditioned on x_0, can be written according to these two cases (Sahoo et al., 2024):

$$q_{\text{cat}}(x_{t'} \mid x_t, x_0) = \begin{cases} \mathbf{1}_{x_{t'} = x_t}, & \text{if } x_t \neq m, \\ \frac{\alpha_{t'} - \alpha_t}{1-\alpha_t}\mathbf{1}_{x_{t'} = x_0} + \frac{1-\alpha_{t'}}{1-\alpha_t}\mathbf{1}_{x_{t'} = m}, & \text{if } x_t = m. \end{cases} \tag{5}$$

Intuitively, in the reverse process, when x_t is masked, the clean value is recovered with probability $\frac{\alpha_{t'} - \alpha_t}{1-\alpha_t}$ before time t', and once unmasked, the value remains fixed until the end ($t = 0$). To learn the reverse unmasking process, Sahoo et al. (2024) propose a reverse process parametrization that matches the form of Equation 5 but replaces (the unknown) x_0 by a network prediction. Importantly, to make this process scalable to a large number of features, we additionally leverage the fact that $x_0 \in \mathcal{V}$ to constrain the network's predictions to the valid categories. Formally, with $\Delta^{K-1} = \{\boldsymbol{\pi} \in [0,1]^K \mid \sum_{i=1}^K \pi_i = 1\}$ denoting the $(K-1)$-simplex, our network $\boldsymbol{\mu}_{\boldsymbol{\theta}}(\cdot, \cdot, s) \colon \mathcal{U} \times [0,1] \to \Delta^{|\mathcal{V}|-1}$ is a schema-conditional classifier over the categories in $\mathcal{V}$. Using the notation $\boldsymbol{\pi}[v]$ to denote the probability assigned by $\boldsymbol{\pi} \in \Delta^{|\mathcal{V}|-1}$ to category $v \in \mathcal{V}$, our proposed reverse-process parametrization is given by

$$p_{\boldsymbol{\theta}}(x_{t'} \mid x_t, s) = \begin{cases} \mathbf{1}_{x_{t'} = x_t}, & \text{if } x_t \neq m, \\ \frac{\alpha_{t'} - \alpha_t}{1-\alpha_t}\boldsymbol{\mu}_{\boldsymbol{\theta}}(x_t, t, s)[x_{t'}]\mathbf{1}_{x_{t'} \in \mathcal{V}} + \frac{1-\alpha_{t'}}{1-\alpha_t}\mathbf{1}_{x_{t'} = m}, & \text{if } x_t = m. \end{cases} \tag{6}$$

Our parametrization can be viewed as a generalization of the SUBS parametrization of Sahoo et al. (2024), which can be recovered by setting $\mathcal{V} = \mathcal{U} \setminus \{m\}$, corresponding to the case where all categorical features share the same domain. Crucially, to evaluate Equation 6, our model needs only to predict probabilities for categories in $\mathcal{V}$ whose size is independent of $\mathcal{U}$ and typically $|\mathcal{V}| \ll |\mathcal{U}|$. This is exactly what makes our process scalable to an arbitrary number of features. Moreover, because our parametrization is independent of the exact definition of $\mathcal{U}$, it natively supports settings in which $\mathcal{U}$ is dynamically updated, such as adapting the model to new categorical distributions. Since we use the same noising process as Sahoo et al. (2024) and our reverse-process parametrization retains the same analytical form as theirs, we can directly use their simplified likelihood bound, which is a weighted cross-entropy loss, to train our model across heterogeneous categorical features,

$$\mathcal{L}_{\text{cat}}(\boldsymbol{\theta}) = \mathbb{E}_{\mathrm{s} \sim p_s} \mathbb{E}_{\mathbf{x}_0 \sim p_0(\cdot \mid s)} \mathbb{E}_{\mathrm{t} \sim \mathcal{U}([0,1])} \mathbb{E}_{\mathbf{x}_t \sim q_{\text{cat}}(\cdot \mid x_0)} \left[\mathbf{1}_{x_t = m} \frac{\alpha_t'}{1-\alpha_t} \log\left(\boldsymbol{\mu}_{\boldsymbol{\theta}}\left(x_t, t, s\right)[x_0]\right) \right]. \tag{7}$$

3.2 Row-level diffusion process

Having defined one-dimensional diffusion processes for numerical and categorical features separately, we can now use them as building blocks to define a row-level mixed-type diffusion process dynamically for any table schema. Here, $\mathbf{x}$ denotes a d-dimensional mixed-type random variable representing the row distribution of a table with schema $\mathcal{S}$. We apply the row-level noising process by applying the feature-level processes independently to each feature, analogously to how pixels are

noised independently in image diffusion models (Song et al., 2020) and how tokens are masked independently in diffusion language models (Sahoo et al., 2024). Since the choice of the feature-level process depends on the feature type, the row-level noising process is conditioned on the schema:

$$q\left(\boldsymbol{x}_t \mid \boldsymbol{x}_0, \mathcal{S}\right) = \prod\nolimits_{j \in \mathcal{J}_{\text{num}}} q_{\text{num}}\left(x_{t,j} \mid x_{0,j}\right) \prod\nolimits_{j \in \mathcal{J}_{\text{cat}}} q_{\text{cat}}\left(x_{t,j} \mid x_{0,j}\right), \tag{8}$$

where q_{num} and q_{cat} are the feature-level noising processes defined in Equations 1 and 4, respectively. To model the row-level reverse process, the network needs to predict the clean values for the numerical features, and the unmasked categories for the (masked) categorical features. Crucially, by processing the full row jointly, the model can capture row-level distributions beyond column marginals. Formally, our final network $\boldsymbol{\mu}_{\boldsymbol{\theta}}(\cdot, \cdot, \mathcal{S})\colon \left(\prod_{j=1}^{d} \mathcal{X}_j\right) \times [0,1] \to \prod_{j=1}^{d} \mathcal{Y}_j$, with

$$\mathcal{X}_j = \begin{cases} \mathbb{R}, & \text{if } j \in \mathcal{J}_{\text{num}}, \\ \mathcal{U}, & \text{if } j \in \mathcal{J}_{\text{cat}}, \end{cases} \qquad \text{and} \qquad \mathcal{Y}_j = \begin{cases} \mathbb{R}, & \text{if } j \in \mathcal{J}_{\text{num}}, \\ \Delta^{|\mathcal{V}_j|-1}, & \text{if } j \in \mathcal{J}_{\text{cat}}, \end{cases} \tag{9}$$

is a schema-conditional row-level denoiser with variable-sized input and output. We can now finally train this network end-to-end on a collection of L tabular datasets via the combined mixed-type loss,

$$\mathcal{L}(\boldsymbol{\theta}) = \mathbb{E}_{\ell \sim p(\{1,2,\ldots,L\})} \mathbb{E}_{\mathbf{x}_0 \sim p_{\text{data}}^{(\ell)}} \mathbb{E}_{\mathfrak{t} \sim \mathcal{U}([0,1])} \mathbb{E}_{\mathbf{x}_t \sim q(\cdot \mid \boldsymbol{x}_0)} \Bigg[\frac{\lambda_t}{|\mathcal{J}_{\text{num}}|} \left\| \boldsymbol{\mu}_{\boldsymbol{\theta}}^{\text{num}}\left(\boldsymbol{x}_t, t, \mathcal{S}\right) - \boldsymbol{x}_0^{\text{num}} \right\|_2^2 + \frac{\alpha_t'}{|\mathcal{J}_{\text{cat}}|(1-\alpha_t)} \sum_{j \in \mathcal{J}_{\text{cat}}, x_{t,j} = m} \log\left(\boldsymbol{\mu}_{\boldsymbol{\theta},j}\left(\boldsymbol{x}_t, t, \mathcal{S}\right)[x_{0,j}]\right) \Bigg]. \tag{10}$$

Sampling. Once trained, we can sample synthetic rows from our model for a target dataset given its schema. First, we initialize numerical features with Gaussian noise at the terminal noise level and categorical features with the mask state. We then iteratively reverse the diffusion process along a discretized time grid. At each step, the trained denoiser predicts clean numerical values and categorical probabilities from the current noisy row, conditioned on the schema and time. These predictions determine an Euler step for the numerical probability-flow ODE (Equation 3) and the categorical reverse transitions (Equation 6) from which we sample. We adopt the mixed-type stochastic sampler of TABDIFF (Shi et al., 2025a), based on the EDM sampler (Karras et al., 2022), which injects additional Gaussian noise and probabilistically remasks categorical features before each denoising update. Finally, we invert the numerical preprocessing to recover the original feature scales.

We provide additional details on the diffusion model in Appendix B.1 and pseudocode for the training and sampling procedure in Algorithms 1 and 2, respectively.

3.3 SCHEMA-CONDITIONAL DENOISER NETWORK ARCHITECTURE

So far, we have developed a framework for training a single diffusion model on heterogeneous tabular datasets by parametrizing the reverse process via a schema-conditional denoiser network. From the diffusion process perspective, the denoiser, $\boldsymbol{\mu}_{\boldsymbol{\theta}}$, needs to have the following key properties: (*i*) It takes as input a *variable-sized mixed-type* (noisy) row and a corresponding schema. (*ii*) By leveraging schema information, it predicts a scalar value for every numerical feature and a categorical distribution over the valid categories for every categorical feature. (*iii*) The output is equivariant to the joint permutation of the input row and schema (see Appendix B.3 for a detailed discussion on permutation equivariance). Guided by these properties, we propose a concrete architecture for $\boldsymbol{\mu}_{\boldsymbol{\theta}}$ consisting of a tokenizer, a transformer encoder, and a detokenizer, visualized in Figure 3. The tokenizer represents each noisy feature as a token, the encoder models cross-feature dependencies, and the detokenizer produces numerical value estimates and categorical probability vectors. All parameters are shared across datasets, and type-specific modules are shared between all columns of that type. Let D denote the token dimension.

Schema Embeddings. Most schema information is typically available in textual form. To obtain a numerical representation of this information, we use a frozen pre-trained text embedding model f_{text} that maps strings to d_{emb}-dimensional vectors. We follow Lin et al. (2025) and use GTE (Li et al., 2023) as the text embedding model. We apply f_{text} to the table's name, column names, and categories, followed by learned linear projection layers to obtain D-dimensional schema embeddings corresponding to each textual schema component

$$\boldsymbol{e}_{\text{name}}, \qquad \left\{\boldsymbol{e}_{\text{col},j}\right\}_{j=1}^{d}, \qquad \left\{\boldsymbol{E}_{\text{cat},j} = \left[\boldsymbol{e}_{\text{cat},j,1}, \ldots, \boldsymbol{e}_{\text{cat},j,K_j}\right]^{\mathsf{T}}\right\}_{j=1}^{d}.$$

Tokenizer. For each noisy feature $x_{t,j}$, we compute a value embedding $\boldsymbol{e}_{\mathrm{val},j} \in \mathbb{R}^D$, depending on its type and whether it is masked or not

$$\boldsymbol{e}_{\mathrm{val},j} = \begin{cases} \mathrm{MLP}_{\mathrm{num,in}}(x_{t,j}), & \text{if } j \in \mathcal{J}_{\mathrm{num}}, \\ \boldsymbol{e}_{\mathrm{cat},j,k}, & \text{if } j \in \mathcal{J}_{\mathrm{cat}} \text{ and } x_{t,j} = v_{j,k}, \\ \mathrm{Attention}\left(\boldsymbol{e}_m^{\mathsf{T}}, \boldsymbol{E}_{\mathrm{cat},j}, \boldsymbol{E}_{\mathrm{cat},j}\right)^{\mathsf{T}}, & \text{if } j \in \mathcal{J}_{\mathrm{cat}} \text{ and } x_{t,j} = m. \end{cases} \tag{11}$$

To obtain the value embeddings, numerical features are processed by an MLP, while unmasked categorical features use their category embeddings. For a masked categorical feature, a learned mask embedding $\boldsymbol{e}_m \in \mathbb{R}^D$ attends to the category embeddings in $\boldsymbol{E}_{\mathrm{cat},j}$, which serve as both keys and values. While $\boldsymbol{e}_m$ is shared across categorical columns and datasets, the attention mechanism's output depends on the current column's vocabulary. We then add type, column, table, and time embeddings to the value embedding to obtain the feature token, $\boldsymbol{h}_j$:

$$\boldsymbol{h}_j = \boldsymbol{e}_{\mathrm{val},j} + \boldsymbol{e}_{\tau_j} + \boldsymbol{e}_{\mathrm{col},j} + \boldsymbol{e}_{\mathrm{name}} + \boldsymbol{e}_{\mathrm{time}}, \tag{12}$$

where $\boldsymbol{e}_{\tau_j} \in \{\boldsymbol{e}_{\mathrm{num}}, \boldsymbol{e}_{\mathrm{cat}}\}$ are learnable type embeddings and $\boldsymbol{e}_{\mathrm{time}}$ is a sinusoidal embedding of the time step t followed by an MLP (Kotelnikov et al., 2023).

Backbone. A transformer encoder (Vaswani et al., 2017) processes the feature tokens jointly to capture cross-column relationships

$$\left[\hat{\boldsymbol{h}}_1, \ldots, \hat{\boldsymbol{h}}_d\right]^{\mathsf{T}} = \mathrm{TransformerEncoder}\left(\left[\boldsymbol{h}_1, \ldots, \boldsymbol{h}_d\right]^{\mathsf{T}}\right) \in \mathbb{R}^{d \times D}. \tag{13}$$

The encoder does *not* use positional encodings. Together with the feature-wise tokenizer and detokenizer, this makes the model equivariant to jointly permuting columns and their corresponding schema entries. Column identities are preserved through the schema embeddings.

Detokenizer. The detokenizer maps the processed token for column j, $\hat{\boldsymbol{h}}_j$, to the final output. For numerical columns, a shared MLP predicts the clean feature value

$$\boldsymbol{\mu}_{\boldsymbol{\theta},j}(\boldsymbol{x}_t, t, \mathcal{S}) = \mathrm{MLP}_{\mathrm{num,out}}(\hat{\boldsymbol{h}}_j) \in \mathbb{R}, \quad j \in \mathcal{J}_{\mathrm{num}}. \tag{14}$$

For categorical columns, a shared single-layer transformer decoder takes the K_j category embeddings as target tokens and the contextualized feature as a single memory token

$$\left[\hat{\boldsymbol{e}}_{\mathrm{cat},j,1}, \ldots, \hat{\boldsymbol{e}}_{\mathrm{cat},j,K_j}\right]^{\mathsf{T}} = \mathrm{TransformerDecoder}\left(\mathrm{target} = \boldsymbol{E}_{\mathrm{cat},j}, \mathrm{memory} = \hat{\boldsymbol{h}}_j^{\mathsf{T}}\right) \in \mathbb{R}^{K_j \times D} \tag{15}$$

Then, a shared scalar projection produces category logits, which are fed into a softmax function,

$$\{z_{j,k} = \mathrm{Lin}_{\mathrm{out}}(\hat{\boldsymbol{e}}_{\mathrm{cat},j,k})\}_{k=1}^{K_j}, \qquad \boldsymbol{\mu}_{\boldsymbol{\theta},j}(\boldsymbol{x}_t, t, \mathcal{S}) = \mathrm{Softmax}\left(\{z_{j,k}\}_{k=1}^{K_j}\right), \quad j \in \mathcal{J}_{\mathrm{cat}}. \tag{16}$$

4 Experiments

Overview. We evaluate CDMD in two complementary settings. First, we study *in-domain cross-dataset generation* (Section 4.1): can a single model trained jointly on multiple tabular datasets generate high-quality synthetic data for each dataset? Second, we study *generation in constrained settings* (Section 4.2): can cross-dataset pre-training improve generation on previously unseen datasets when either the amount of target training data or the number of training epochs is limited? We begin by describing the experimental setup shared by both evaluation settings with additional implementation details in Appendix C.4.

Evaluation datasets. We use seven real-world tabular datasets widely adopted in tabular generation works (Zhang et al., 2024; Shi et al., 2025a) as our target datasets for evaluation: *Adult*, *Default*, *Magic*, *Shoppers*, *Diabetes*, *Beijing*, and *News*. They span diverse domains, contain different combinations of numerical and categorical features, and each comes with an inherent machine learning task: binary classification, multiclass classification, or regression. We split each dataset into two equally sized subsets: a training set for fitting and developing the generative models, and a held-out test set for evaluating synthetic data quality. Detailed dataset information is in Appendix C.2.

Evaluation metrics. Following Zhang et al. (2024); Shi et al. (2025a), we assess synthetic data quality using seven metrics across three evaluation axes (DataCebo, 2026; Alaa et al., 2022).

Table 1: In-domain cross-dataset generation results (Section 4.1). Entries report mean ± standard deviation over three runs. Average is the mean over datasets. TRAIN SET is the real-data reference. Bold and underlined entries mark the best and second-best scores in each column, excluding TRAIN SET. GREAT cannot be applied to News because of its maximum length limit.

Generation Quality: $Q = (F + U)/2$ (↑)

	Method \ Dataset	Adult	Default	Magic	Shoppers	Diabetes	Beijing	News	Average
	TRAIN SET	89.64 ±0.00	82.52 ±0.00	89.69 ±0.00	89.57 ±0.00	77.40 ±0.00	88.15 ±0.00	87.47 ±0.00	86.35
Single-dataset	SMOTE	86.17 ±0.03	79.33 ±0.05	88.40 ±0.12	86.73 ±0.06	72.41 ±0.05	**86.72** ±0.03	85.66 ±0.38	83.63
	CTGAN	78.06 ±1.43	68.44 ±0.41	74.56 ±1.35	74.69 ±2.00	59.96 ±2.40	81.28 ±0.51	77.64 ±0.38	73.52
	TVAE	82.47 ±0.84	74.15 ±0.56	80.26 ±1.37	72.92 ±0.50	60.53 ±1.27	81.61 ±0.96	80.45 ±0.27	76.06
	GREAT	82.64 ±0.21	76.54 ±0.36	82.04 ±0.13	83.46 ±0.18	70.32 ±0.14	84.14 ±0.09	–	79.86
	TABDDPM	87.76 ±0.04	79.85 ±0.38	88.33 ±0.10	85.40 ±0.12	41.48 ±0.18	86.00 ±0.31	38.59 ±5.89	72.49
	TABSYN	88.43 ±0.07	81.36 ±0.12	88.85 ±0.00	87.84 ±0.30	72.33 ±0.08	86.18 ±0.11	84.87 ±0.16	84.27
	TABDIFF	88.28 ±0.10	81.39 ±0.23	88.67 ±0.06	87.93 ±0.25	75.38 ±0.17	86.15 ±0.30	81.75 ±0.11	84.22
Cross-dataset	TABEBM	62.40 ±0.60	57.41 ±0.96	84.25 ±0.17	70.00 ±0.56	43.46 ±0.37	77.31 ±0.11	67.66 ±5.85	66.07
	CTSYN	67.66 ±0.62	66.80 ±0.79	72.46 ±0.35	63.67 ±1.71	48.04 ±0.04	76.07 ±0.05	79.66 ±0.09	67.76
	CDMD (OURS)	**88.59** ±0.03	**81.68** ±0.07	**88.92** ±0.08	**88.48** ±0.05	**76.09** ±0.05	85.90 ±0.27	**86.00** ±0.03	**85.09**

Overall Evaluation Score: $S = (F + U + P)/3$ (↑)

	Method \ Dataset	Adult	Default	Magic	Shoppers	Diabetes	Beijing	News	Average
	TRAIN SET	59.79 ±0.00	55.05 ±0.00	59.99 ±0.00	60.09 ±0.00	51.60 ±0.00	58.77 ±0.00	58.31 ±0.00	57.66
Single-dataset	SMOTE	65.30 ±0.04	60.79 ±0.03	64.60 ±0.10	63.46 ±0.18	52.47 ±0.01	66.62 ±0.04	61.89 ±0.22	62.16
	CTGAN	78.34 ±0.87	72.21 ±0.35	78.93 ±0.93	76.57 ±1.38	68.88 ±1.27	82.85 ±0.29	80.26 ±0.33	76.86
	TVAE	80.80 ±0.40	76.69 ±0.05	80.64 ±0.53	77.63 ±0.39	67.80 ±0.69	82.60 ±0.41	80.73 ±0.12	78.13
	GREAT	79.49 ±0.19	76.07 ±0.39	80.31 ±0.14	80.98 ±0.28	71.62 ±0.07	80.51 ±0.05	–	78.16
	TABDDPM	83.34 ±0.11	78.13 ±0.26	83.18 ±0.18	78.64 ±0.18	60.53 ±0.08	**84.43** ±0.25	59.06 ±3.93	75.33
	TABSYN	83.61 ±0.07	79.00 ±0.02	83.99 ±0.10	83.30 ±0.41	74.38 ±0.09	84.32 ±0.09	82.38 ±0.08	81.57
	TABDIFF	81.15 ±0.02	76.56 ±0.05	83.65 ±0.11	**83.43** ±0.15	72.80 ±0.09	84.29 ±0.28	79.43 ±0.05	80.19
Cross-dataset	TABEBM	69.55 ±0.52	67.99 ±0.47	69.70 ±0.16	71.96 ±0.45	60.12 ±0.25	78.97 ±0.15	62.98 ±3.99	68.75
	CTSYN	73.26 ±0.41	71.72 ±0.44	78.07 ±0.07	69.54 ±1.18	62.81 ±0.04	82.86 ±0.05	79.60 ±0.12	73.98
	CDMD (OURS)	**83.77** ±0.19	**79.12** ±0.24	**83.99** ±0.19	83.12 ±0.04	**75.84** ±0.04	84.08 ±0.22	**82.45** ±0.05	**81.77**

(1) *Distributional Fidelity* (F): *Shape* and *Trend* compare real and synthetic column marginals and pairwise column correlations, respectively, whereas α-*Precision* and β-*Recall* assess sample-level resemblance to the real distribution and its coverage, respectively. We use the held-out test set as the reference data for these metrics and define the fidelity score, F, as their average. **(2)** *Predictive Utility* (U): *Machine Learning Efficacy (MLE)* measures how well synthetic data preserves information useful for the dataset's prediction task. Following Zhang et al. (2024), we train an XGBoost classifier or regressor on synthetic data and evaluate it on the held-out test data. The utility score, U, is based on AUC and normalized RMSE for classification and regression datasets, respectively. **(3)** *Privacy proxies* (P): We use a *distance-to-closest-record (DCR)* score to assess whether synthetic rows are disproportionately close to the training set relative to the equally sized test set and *Authenticity* to assess potential copying through each synthetic row's proximity to its nearest training record relative to local training-data distances. These metrics provide empirical diagnostics of potential memorization, and their mean defines the privacy proxy score P. We express all individual metrics on a 0 to 100 scale, with higher values indicating better performance. We provide detailed metric computation and normalization in Appendix C.1. Due to space constraints, we report aggregated scores for our experiments in the main text and provide detailed results in Appendix D. We aggregate the Fidelity and Utility scores into a *Generation Quality* score (Q) and the three axes' scores into an *Overall Evaluation Score* (S). Averaging first within each axis and then across axes gives each evaluation dimension equal weight, independently of the number of metrics used to measure it. Formally,

$$Q = (F + U)/2 \quad (\textit{Generation Quality}), \qquad S = (F + U + P)/3 \quad (\textit{Overall Evaluation Score}).$$

Baselines. We compare CDMD against nine tabular generation methods, categorized as single-dataset and cross-dataset methods. **(1)** *Single-dataset methods*. We select representative methods from different model families: (i) *interpolation-based*, SMOTE (Chawla et al., 2002); (ii) *VAE-based*, TVAE (Xu et al., 2019); (iii) *GAN-based*, CTGAN (Xu et al., 2019); (iv) *language model-based*, GREAT (Borisov et al., 2022b); and (v) *diffusion-based*, TABDDPM (Kotelnikov et al., 2023), TABSYN (Zhang et al., 2024), and TABDIFF (Shi et al., 2025a). **(2)** *Cross-dataset methods*. We evaluate TABEBM (Margeloiu et al., 2024), an energy-based method built on a pretrained in-context classifier, TABPFN (Hollmann et al., 2022), and CTSYN (Lin et al., 2025), a schema-conditional cross-dataset generator. At the time of writing, we could not locate a publicly available implementation of LATABLE (van Breugel et al., 2024), so we exclude it from our evaluations.

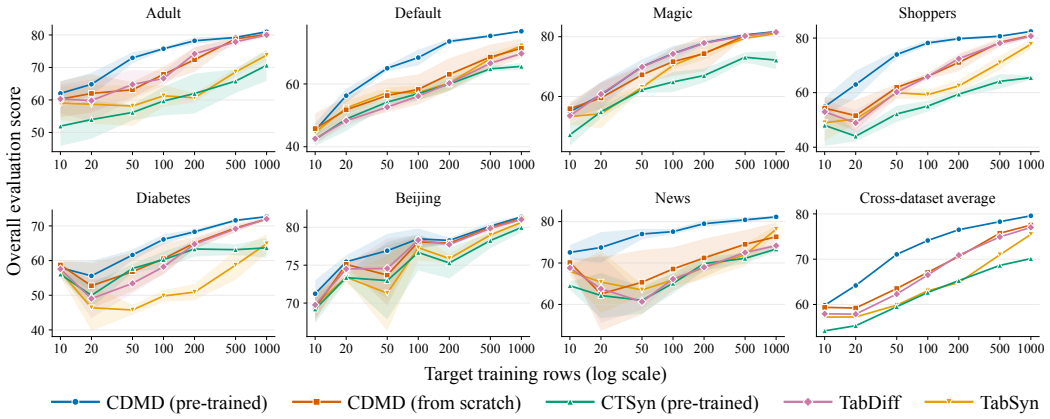

Figure 1: Sample-constrained generation results: Overall Evaluation Score (S) vs. the number of target training rows. Lines and shaded regions show the mean ± standard deviation over five runs using independently-sampled target training subsets. The final panel shows the mean across datasets.

4.1 In-domain cross-dataset generation

This experiment tests whether a single CDMD checkpoint can generate high-quality synthetic data across multiple datasets. To this end, we train CDMD jointly on the training splits of the seven evaluation datasets and use the resulting checkpoint to generate synthetic data for each dataset conditioned on its schema. We follow the same training protocol for CTSyn, while TabEBM uses the corresponding training data as context, and the single-dataset baselines are applied separately to each dataset's training split. We then evaluate all methods using the same evaluation protocol.

Table 1 shows that CDMD achieves the highest average generation quality score, 85.09, followed by TabSyn (84.27) and TabDiff (84.22), and obtains the best generation-quality score on six out of seven datasets. It also achieves the highest average overall score, 81.77, followed by TabSyn at 81.57, and substantially outperforms jointly trained CTSyn on both scores. The detailed results in Appendix D.1 clarify this advantage: CDMD has higher average distributional fidelity than TabSyn and TabDiff, while matching TabDiff's average predictive utility. Notably, CDMD achieves this performance while using a fraction of the total trainable parameters compared to the best single-dataset baselines: TabSyn and TabDiff both have 74.8M total trainable parameters across the seven datasets, while CDMD has 10.7M parameters (excluding the frozen text embedding model, which is used once to get schema embeddings and is not needed during training nor sampling). These results demonstrate that a single CDMD checkpoint can match or exceed strong, separately trained generators while supporting heterogeneous schemas.

4.2 Generation in constrained settings

This experiment tests whether cross-dataset transfer can improve generation quality when training data or epochs are limited. Following van Breugel et al. (2024) and Lin et al. (2025), we adopt a pre-training-and-fine-tuning setup. We pre-train CDMD (a different instance from the previous experiment) on a corpus of 337 tabular datasets, filtered by Yan et al. (2024). We exclude the evaluation datasets from this corpus through automated checks followed by manual verification. Further corpus details are in Appendix C.3. Starting from the same pretrained checkpoint for each run, we fine-tune CDMD separately on each evaluation dataset, using either a subset of the training set or the full training set for a limited number of epochs. We compare against CTSyn pretrained on the same corpus, TabSyn and TabDiff as the strongest single-dataset baselines from the previous experiment, and CDMD trained from scratch. In the data-constrained setting, all methods use the same target training rows, and in the epoch-constrained setting, they use the same number of epochs.

Adaptation with limited target data. For each evaluation dataset and sample budget $N \in \{10, 20, 50, 100, 200, 500, 1000\}$, we draw five independent random subsets of N rows from the training split to account for variability in target-data selection. Within each draw, all methods, including numerical pre-processing, are fitted on exactly the same rows. For each run, we generate a

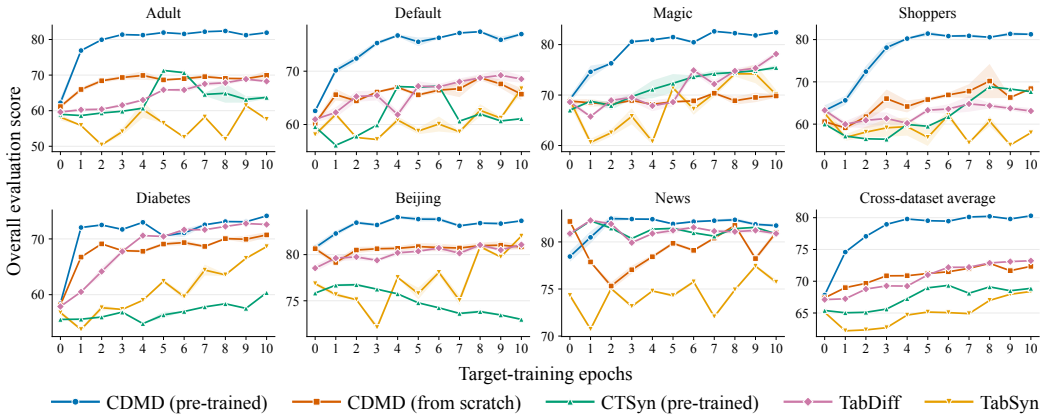

Figure 2: Epoch-constrained generation results: Overall Evaluation Score (S) vs. training epochs. Lines and shaded regions indicate the mean $\pm$ standard deviation over three runs.

synthetic dataset matching the size of the held-out test set and compute the evaluation scores (for DCR scores, we subsample the held-out set to N and report the average over 20 seeds). Keeping the synthetic sample count fixed across budgets controls for its effect on the metrics and maintains consistency with the previous experiment. Figure 1 shows that pre-trained CDMD achieves the highest cross-dataset-average overall score at every sample budget, with the clearest advantage at tens to a few hundred target examples. Improvements are particularly evident on Adult, Default, Shoppers, and News. As more target rows become available, the average gap generally narrows as models trained from scratch improve. The generation-quality results in Appendix D.2 show that the benefit also appears when privacy proxies are excluded from the aggregate. These results demonstrate that cross-dataset pretraining supports effective generation from limited target data.

Adaptation with limited training epochs. We next train or adapt each method on the full training split of each target dataset and evaluate generation quality over the first ten epochs, with epoch 0 denoting no adaptation/training. An epoch corresponds to one complete pass through the training rows (by both components for two-stage models), regardless of batch size. This setting assesses how pretraining affects generation quality during the early stages of training on a new dataset. As shown in Figure 2, pretrained CDMD improves rapidly during the initial epochs and approaches its observed plateau within a few epochs. It attains the highest cross-dataset average overall score after one epoch and retains this advantage through epoch ten, with clear improvements on Adult, Default, Magic, and Shoppers. The advantage over CDMD trained from scratch supports the benefit of pretrained adaptation within a limited number of target-training epochs. Appendix D.3 shows the corresponding generation-quality curves, which support the same conclusion without the privacy proxies. Together with the limited-data experiment, these results establish the value of pretrained CDMD under both constraints: scarce target observations and limited adaptation epochs.

5 Conclusion

In this work, we introduced CDMD, a mixed-type diffusion model that can be trained end-to-end across heterogeneous tabular datasets. By combining numerical and categorical diffusion with a shared schema-conditioned denoiser, CDMD accommodates varying feature sets and categorical vocabularies directly in the mixed-type feature space. Experiments on seven real-world datasets show that a single jointly trained checkpoint matches or outperforms strong separately trained generators on average. Pretraining on 337 datasets further improves generation on unseen datasets under limited target data and adaptation epochs.

Limitations and future work. Our current adaptation strategy relies on fine-tuning, requiring additional optimization for each target dataset. Future work could extend CDMD to in-context generation, enabling adaptation by conditioning on example rows without updating model parameters. Our experiments are also limited in scale, leaving performance at larger scales an open question. Nevertheless, the shared schema-conditional architecture provides a starting point for scaling tabular generative models to larger and more diverse corpora.

AI USE STATEMENT

In this work, we used generative AI tools to provide feedback on research methodology and experiments and help interpret their results. We have not used generative AI tools to generate synthetic data sets, help develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, assist in the writing of proofs, propose or refine hypotheses, assist with translation, clean and reformat a dataset, implement methods, or support qualitative and thematic data analysis. Additionally, we used generative AI tools to create or modify scientific figures or images, summarize or analyze existing literature, edit writing, and propose titles for figures and tables. We have reviewed all AI-assisted work. We checked and edited all LLM-generated text and code. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We describe the proposed model and experimental protocols in the main text and provide supporting details in the appendix. Appendix B specifies the diffusion processes, denoiser architecture, and training and sampling algorithms. Appendix C documents dataset sources, splits, and preprocessing; pretraining corpus construction; baseline implementations; model configurations; optimization and adaptation procedures; and evaluation metrics. Appendix D provides per-dataset and component-level results that support the aggregate comparisons.

REFERENCES

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

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

Vadim Borisov, Tobias Leemann, Kathrin Seßler, Johannes Haug, Martin Pawelczyk, and Gjergji Kasneci. Deep neural networks and tabular data: A survey. *IEEE transactions on neural networks and learning systems*, 35(6):7499–7519, 2022a.

Vadim Borisov, Kathrin Seßler, Tobias Leemann, Martin Pawelczyk, and Gjergji Kasneci. Language models are realistic tabular data generators. *arXiv preprint arXiv:2210.06280*, 2022b.

Donghong Cai, Jiarui Feng, Yanbo Wang, Da Zheng, Yixin Chen, and Muhan Zhang. Tabdlm: Free-form tabular data generation via joint numerical-language diffusion. *arXiv preprint arXiv:2602.22586*, 2026.

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

Inc. DataCebo. *Synthetic Data Metrics*. DataCebo, Inc., 02 2026. URL `https://docs.sdv.dev/sdmetrics/`. Version 0.27.1.

Xinyan Han, Yan Lu, Xiaoyu Lin, Yuanyuan Jiang, Yuanrui Wang, Xuanyue Li, Wenchao Zou, and Xingxuan Zhang. Breaking the quality-privacy tradeoff in tabular data generation via in-context learning. *arXiv preprint arXiv:2605.04911*, 2026.

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

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

Xiangjian Jiang, Mingxuan Liu, Nikola Simidjievski, Tassilo Klein, and Mateja Jamnik. Tabular foundation model for generative modelling. *arXiv preprint arXiv:2605.09424*, 2026.

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

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

Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. Towards general text embeddings with multi-stage contrastive learning. *arXiv preprint arXiv:2308.03281*, 2023.

Xiaofeng Lin, Chenheng Xu, Matthew Yang, and Guang Cheng. Ctsyn: A foundation model for cross tabular data generation. In *International Conference on Learning Representations*, volume 2025, pp. 47938–47959, 2025.

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

Andrei Margeloiu, Xiangjian Jiang, Nikola Simidjievski, and Mateja Jamnik. Tabebm: A tabular data augmentation method with distinct class-specific energy-based models. *Advances in Neural Information Processing Systems*, 37:72094–72144, 2024.

Zhaozhi Qian, Bogdan-Constantin Cebere, and Mihaela van der Schaar. Synthcity: facilitating innovative use cases of synthetic data in different data modalities. *arXiv preprint arXiv:2301.07573*, 2023.

Jingang Qu, David Holzmüller, Gaël Varoquaux, and Marine Le Morvan. Tabicl: A tabular foundation model for in-context learning on large data. *arXiv preprint arXiv:2502.05564*, 2025.

Subham S Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin T Chiu, Alexander Rush, and Volodymyr Kuleshov. Simple and effective masked diffusion language models. *Advances in Neural Information Processing Systems*, 37:130136–130184, 2024.

Juntong Shi, Minkai Xu, Harper Hua, Hengrui Zhang, Stefano Ermon, and Jure Leskovec. Tabdiff: a mixed-type diffusion model for tabular data generation. In *International Conference on Learning Representations*, volume 2025, pp. 37353–37375, 2025a.

Ruxue Shi, Yili Wang, Mengnan Du, Xu Shen, Yi Chang, and Xin Wang. A comprehensive survey of synthetic tabular data generation. *arXiv preprint arXiv:2504.16506*, 2025b.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pp. 2256–2265. pmlr, 2015.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

Mihaela Catalina Stoian, Eleonora Giunchiglia, and Thomas Lukasiewicz. A survey on deep learning approaches for tabular data generation: Utility, alignment, fidelity, privacy, diversity, and beyond. *arXiv preprint arXiv:2503.05954*, 2025.

Boris Van Breugel and Mihaela Van der Schaar. Beyond privacy: Navigating the opportunities and challenges of synthetic data. *arXiv preprint arXiv:2304.03722*, 2023.

Boris Van Breugel and Mihaela Van Der Schaar. Why tabular foundation models should be a research priority. *arXiv preprint arXiv:2405.01147*, 2024.

Boris van Breugel, Jonathan Crabbé, Rob Davis, and Mihaela van der Schaar. Latable: towards large tabular models. *arXiv preprint arXiv:2406.17673*, 2024.

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

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

Jiahuan Yan, Bo Zheng, Hongxia Xu, Yiheng Zhu, Danny Chen, Jimeng Sun, Jian Wu, and Jintai Chen. Making pre-trained language models great on tabular prediction. In *International Conference on Learning Representations*, volume 2024, pp. 23570–23593, 2024.

Chao Ye, Guoshan Lu, Haobo Wang, Liyao Li, Sai Wu, Gang Chen, and Junbo Zhao. Towards cross-table masked pretraining for web data mining. In *Proceedings of the ACM Web Conference 2024*, pp. 4449–4459, 2024.

Hengrui Zhang, Jiani Zhang, Zhengyuan Shen, Balasubramaniam Srinivasan, Xiao Qin, Christos Faloutsos, Huzefa Rangwala, and George Karypis. Mixed-type tabular data synthesis with score-based diffusion in latent space. In *International Conference on Learning Representations*, volume 2024, pp. 52829–52857, 2024.

Xingxuan Zhang, Gang Ren, Han Yu, Hao Yuan, Hui Wang, Jiansheng Li, Jiayun Wu, Lang Mo, Li Mao, Mingchao Hao, et al. Limix: Unleashing structured-data modeling capability for generalist intelligence. *arXiv preprint arXiv:2509.03505*, 2025.

A Related work

Single-dataset tabular generative models. Tabular data generation must accommodate substantial heterogeneity across columns within a single dataset: features differ in data type, numerical scale, marginal distribution, and categorical vocabulary, while their dependencies determine the joint row distribution. Neural approaches such as TVAE and CTGAN (Xu et al., 2019) address mixed numerical and categorical data through dedicated representations; CTGAN additionally uses mode-specific normalization and conditional sampling to handle multimodal numerical distributions and imbalanced categories. Diffusion models offer several approaches to this heterogeneity. TabDDPM (Kotelnikov et al., 2023) combines Gaussian diffusion for numerical features with multinomial diffusion for categorical features, using a joint denoiser to capture dependencies. TabSyn (Zhang et al., 2024) instead learns a Transformer-based variational autoencoder and applies score-based diffusion in its continuous latent space. Most closely related to our mixed-type formulation, TabDiff (Shi et al., 2025a) jointly models numerical and categorical features using Gaussian and masked diffusion models, with learnable, feature-specific schedules that accommodate differences in column distributions. TabDLM (Cai et al., 2026) further combines numerical diffusion with masked diffusion over categorical and free-form text tokens using a pretrained diffusion language model. These methods are fitted to individual datasets in their reported generative experiments. CDMD addresses the additional challenge of sharing a mixed-type generator across datasets with different schemas and categorical vocabularies.

Cross-dataset tabular generative models. Cross-dataset generators seek to transfer generative knowledge across tables despite differences in feature semantics, types, and dimensionality. LaTable (van Breugel et al., 2024) uses a shared Transformer conditioned on textual dataset and column information. It applies Gaussian diffusion to numerical values and language-model embeddings of categorical values, recovering categorical probabilities through similarity scores against each feature's vocabulary. CTSyn (Lin et al., 2025) learns an autoencoder that maps heterogeneous rows to a common latent representation, together with a schema-guided decoder that reconstructs numerical and categorical values. A schema-conditioned latent diffusion model generates these representations, supporting pretraining across tables and subsequent adaptation. DiffICL (Han et al., 2026) adopts an in-context formulation: a frozen LimiX (Zhang et al., 2025) encoder represents target-table examples, and a pretrained latent diffusion model generates new representations conditioned on context rows. Its reconstruction stage still requires dataset-specific training, fitting a separate lightweight decoder for each feature using target-table data. Thus, although the diffusion model is shared, the complete generation pipeline includes newly trained target-specific components. These approaches perform diffusion in continuous representations. CDMD defines diffusion directly over mixed numerical and categorical states, with schema-conditioned modules shared across datasets and categorical predictions restricted to each feature's vocabulary.

Tabular foundation models and their generative adaptations. Predictive tabular foundation models, including TabPFN (Hollmann et al., 2022; 2025) and TabICL (Qu et al., 2025), learn transferable prediction procedures through pretraining across (synthetic) tasks. They use labeled examples from a target dataset as context to predict unseen targets without updating model parameters. Their standard objective is conditional target prediction, while synthetic row generation requires a model of the joint distribution over features. Hollmann et al. (2025) show that TabPFN can nevertheless serve as an autoregressive generator by treating successive columns as prediction targets and sampling each feature conditional on previously generated features and real-data context. This construction reuses pretrained conditional predictors and introduces feature ordering into the generation process. Other adaptations construct explicit generative models from pretrained predictors. TabPFGen (Ma et al., 2024) derives a class-conditional energy from frozen TabPFN outputs and generates samples using stochastic gradient Langevin dynamics. TabEBM (Margeloiu et al., 2024) constructs distinct class-specific energy models through surrogate binary classification tasks evaluated by a pretrained in-context classifier. TabFORGE (Jiang et al., 2026) uses frozen representations from a pretrained TabPFN encoder, pretrains a latent diffusion Transformer, and subsequently trains a decoder on denoised representations. Its reconstruction interface contains table-specific detokenizers with feature-specific output heads; adaptation initializes new heads and fine-tunes the pretrained diffusion and decoder networks. CDMD learns its generative model directly in the mixed-type feature space, using shared schema-conditioned input and output modules without relying on a pretrained tabular predictor.

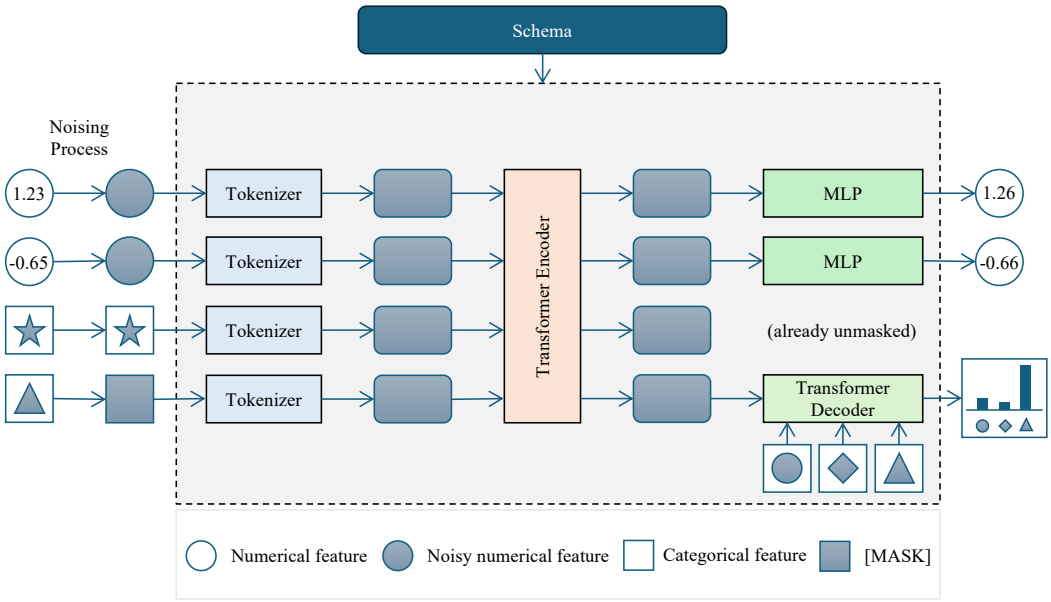

Figure 3: Overview of CDMD. Numerical features are corrupted with Gaussian noise, while categorical features are randomly masked. The tokenizer combines each noisy feature with schema information, and a shared transformer encoder captures dependencies across columns. A shared MLP predicts clean numerical values, while a shared transformer decoder uses category embeddings and the contextualized feature token to produce probabilities over each categorical column's valid vocabulary. These predictions guide the mixed-type reverse diffusion process. The schema determines the number of feature tokens and categorical outputs, while parameters are shared across datasets and columns of the same type, allowing a single model to handle heterogeneous tables.

B Method Details

Algorithm 1 Cross-dataset training of CDMD

Require: L datasets $\{\mathcal{T}^{(\ell)}\}_{\ell=1}^{L} = \{(\mathcal{S}^{(\ell)}, \boldsymbol{X}^{(\ell)})\}_{\ell=1}^{L}$, denoising network $\boldsymbol{\mu}_{\boldsymbol{\theta}}$

repeat
- Sample $\ell \sim p_{\text{dataset}}(\{1, 2, \ldots, L\})$ ▷ Sample dataset index
- Sample $\boldsymbol{x}_0 \sim \mathcal{U}(\boldsymbol{X}^{(\ell)})$ ▷ Sample clean row
- Sample $t \sim \mathcal{U}([0, 1])$ ▷ Sample diffusion time
- ▷ Forward noising process:
- Initialize $\boldsymbol{x}_t$ with same shape as $\boldsymbol{x}_0$
- $\boldsymbol{x}_t^{\text{num}} \leftarrow \boldsymbol{x}_0^{\text{num}} + \sigma_t \boldsymbol{\epsilon}$ with $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ ▷ (Equation 1)
- $x_{t,j} \sim q_{\text{cat}}(x_{t,j} \mid x_{0,j}) \quad \forall j \in \mathcal{J}_{\text{cat}}^{(\ell)}$ ▷ Mask with probability $1 - \alpha_t$ (Equation 4)
- ▷ Network prediction:
- $\hat{\boldsymbol{x}} \leftarrow \boldsymbol{\mu}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t, \mathcal{S}^{(\ell)})$
- $\mathcal{L}_{\text{num}} \leftarrow \frac{\lambda_t}{|\mathcal{J}_{\text{num}}^{(\ell)}|} \|\hat{\boldsymbol{x}}^{\text{num}} - \boldsymbol{x}_0^{\text{num}}\|_2^2$
- $\mathcal{L}_{\text{cat}} \leftarrow \frac{\alpha_t'}{|\mathcal{J}_{\text{cat}}^{(\ell)}|(1-\alpha_t)} \sum_{j \in \mathcal{J}_{\text{cat}}^{(\ell)}, x_{t,j}=m} \log\left(\hat{\boldsymbol{x}}_j[x_{0,j}]\right)$
- Take gradient descent step on $\nabla_{\boldsymbol{\theta}}(\mathcal{L}_{\text{num}} + \mathcal{L}_{\text{cat}})$

until converged.

Algorithm 2 Schema-conditioned sampling from CDMD

Require: Schema $\mathcal{S}$, trained denoising network $\boldsymbol{\mu}_{\boldsymbol{\theta}}$, discretization time steps $\{t_i : i \in \{0, 1, \ldots, N\}\}$ such that $t_0 = 0 < t_1 < \cdots < t_N = 1$

$\boldsymbol{x}_1^{\text{num}} \sim \mathcal{N}(\mathbf{0}_{|\mathcal{J}_{\text{num}}|}, \sigma_{\max}^2 \boldsymbol{I}_{|\mathcal{J}_{\text{num}}|})$
$\forall j \in \mathcal{J}_{\text{cat}}\ x_{1,j} \leftarrow m$
for $i \in \{N, N-1, \ldots, 1\}$ **do**
  if $i < N$ **then**
    $\hat{t}_i \leftarrow \min(\sigma^{-1}(\sigma_{t_i} + \frac{1}{N}\sigma_{t_i}), 1)$ ▷ Select temporarily increased noise level $\hat{t}_i$
    ▷ Numerical forward perturbation
    $\boldsymbol{x}_{\hat{t}_i}^{\text{num}} \leftarrow \boldsymbol{x}_{t_i}^{\text{num}} + \sqrt{\sigma_{\hat{t}_i}^2 - \sigma_{t_i}^2}\boldsymbol{\epsilon}_i$ with $\boldsymbol{\epsilon}_i \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$
    ▷ Categorical forward perturbation
    $\forall j \in \mathcal{J}_{\text{cat}}\ x_{\hat{t}_i,j} \sim q_{\text{cat}}(x_{\hat{t}_i,j} \mid x_{t_i,j})$ ▷ (Equation 20)
  else
    $\hat{t}_i = t_i$ ▷ Stochastic sampler is disabled at the first sampling iteration
  end if
  ▷ Model prediction
  $\hat{\boldsymbol{x}}_i \leftarrow \boldsymbol{\mu}_{\boldsymbol{\theta}}(\boldsymbol{x}_{\hat{t}_i}, \hat{t}_i, \mathcal{S})$
  ▷ Numerical backward sampling
  $d_i \leftarrow \frac{\sigma'_{\hat{t}_i}}{\sigma_{\hat{t}_i}}(\boldsymbol{x}_{\hat{t}_i}^{\text{num}} - \hat{\boldsymbol{x}}_i^{\text{num}})$ ▷ Evaluate $\frac{d\boldsymbol{x}^{\text{num}}}{dt}$ at $\hat{t}_i$
  $\boldsymbol{x}_{t_{i-1}}^{\text{num}} \leftarrow \boldsymbol{x}_{\hat{t}_i}^{\text{num}} + (t_{i-1} - \hat{t}_i)d_i$ ▷ Take Euler step from $\hat{t}_i$ to t_{i-1}
  ▷ Categorical backward sampling
  $\forall j \in \mathcal{J}_{\text{cat}}\ x_{t_{i-1},j} \sim p_{\boldsymbol{\theta}}(x_{t_{i-1},j} \mid x_{\hat{t}_i,j}, \hat{\boldsymbol{x}}_{i,j})$ ▷ (Equation 21)
end for
return $\boldsymbol{x}_{t_0}$

B.1 Diffusion model details

Diffusion schedules. The Gaussian diffusion schedule $\sigma_t \colon [0,1] \to \mathbb{R}_+$ is an increasing function of t. Following Karras et al. (2022) and Shi et al. (2025a), we adopt the following schedule

$$\sigma_t = \left(\sigma_{\min}^{\frac{1}{\rho}} + t\left(\sigma_{\max}^{\frac{1}{\rho}} - \sigma_{\min}^{\frac{1}{\rho}}\right)\right)^{\rho}, \tag{17}$$

and we use the default values in Karras et al. (2022) throughout all our experiments: $\sigma_{\min} = 0.002$, $\sigma_{\max} = 80$ and $\rho = 7$.

The masking diffusion schedule $\alpha_t : [0,1] \to [0,1]$ is a strictly decreasing function of t. We use a simple linear schedule (Austin et al., 2021; Sahoo et al., 2024)

$$\alpha_t = 1 - t. \tag{18}$$

Numerical loss weighting. The numerical loss weighting λ_t is defined as in Karras et al. (2022)

$$\lambda_t = \frac{\sigma_t^2 + \sigma_{\text{data}}^2}{(\sigma_t \sigma_{\text{data}})^2}. \tag{19}$$

Additional masked diffusion details. The step transition induced by the masking forward process from $t' < t$ to t can be written as (Sahoo et al., 2024)

$$q_{\text{cat}}(x_t \mid x_{t'}) = \frac{\alpha_t}{\alpha_{t'}}\mathbf{1}_{x_t = x_{t'}} + \left(1 - \frac{\alpha_t}{\alpha_{t'}}\right)\mathbf{1}_{x_t = m} \tag{20}$$

The row-conditioned reverse categorical process for a column j can be written as

$$p_{\boldsymbol{\theta}}(x_{t'} \mid x_t, \hat{\boldsymbol{x}}) = \begin{cases} \mathbf{1}_{x_{t'} = x_t}, & \text{if } x_t \neq m, \\ \frac{\alpha_{t'} - \alpha_t}{1 - \alpha_t}\hat{\boldsymbol{x}}[x_{t'}]\mathbf{1}_{x_{t'} \in \mathcal{V}} + \frac{1 - \alpha_{t'}}{1 - \alpha_t}\mathbf{1}_{x_{t'} = m}, & \text{if } x_t = m. \end{cases} \tag{21}$$

Training dataset sampling distribution. When training on L datasets, we sample dataset ℓ with probability

$$p_{\text{dataset}}(\ell) = \frac{N^{(\ell)^{\tau_{\text{data}}}}}{\sum_{\ell'=1}^{L} N^{(\ell')^{\tau_{\text{data}}}}} \tag{22}$$

where $N^{(\ell)}$ is the number of training samples and τ_{data} controls the influence of dataset size. In the cross-dataset experiments, we set $\tau_{\text{data}} = 0$, assigning equal sampling probability to all datasets. During pre-training, we use $\tau_{\text{data}} = 0.5$ to accommodate the substantial variation in dataset sizes while limiting the dominance of larger datasets.

Training and sampling. We present pseudocode for training and sampling from CDMD in Algorithms 1 and 2, respectively. Note that Algorithm 1 and Equation 10 assume that each training dataset contains at least one feature of each type. In our implementation, if a dataset contains no features of a given type, we set the corresponding loss to 0.

B.2 Architecture details

We use the same model architecture across all experiments. The text embedding dimension is $d_{\text{emb}} = 768$, and the token dimension is $D = 384$. The Transformer encoder consists of six layers, each with eight attention heads and a feedforward dimension of 768. All MLPs have two layers, with dimensions matched to their respective inputs and outputs. Following TabDiff (Shi et al., 2025a), we adopt EDM network preconditioning for numerical denoising (Karras et al., 2022) using their default hyperparameters. All components share parameters across datasets and across columns of the corresponding type. The model has 10,655,234 trainable parameters, excluding the frozen text embedding model.

B.3 Desiderata for cross-dataset tabular generative models

Ideally, a cross-dataset tabular generative model $p_{\boldsymbol{\theta}}$ should satisfy the following desiderata:

Desiderata for Cross-Dataset Tabular Generative Models

For an arbitrary table schema $\mathcal{S}$, $p_{\boldsymbol{\theta}}(\cdot \mid \mathcal{S})$ should satisfy the following desiderata:

1. **Schema-Valid Support**: Rows not consistent with the schema have zero probability,
$$\forall \boldsymbol{x} \notin \mathcal{X}_{\mathcal{S}},\ p_{\boldsymbol{\theta}}(\boldsymbol{x} \mid \mathcal{S}) = 0.$$
2. **Column Permutation Invariance**: For any permutation function π over the columns, let $\boldsymbol{x}_\pi$ and $\mathcal{S}_\pi$ denote the permuted row and schema, respectively. Then, the probability of a given row remains invariant to column permutation,
$$\forall \boldsymbol{x} \in \mathcal{X}_{\mathcal{S}},\ p_{\boldsymbol{\theta}}(\boldsymbol{x}_\pi | \mathcal{S}_\pi) = p_{\boldsymbol{\theta}}(\boldsymbol{x} \mid \mathcal{S}).$$

CDMD satisfies both desiderata by construction. Schema-valid support follows from the type-specific reverse processes: numerical values remain in $\mathbb{R}$, while categorical predictions are restricted to each column's valid vocabulary, with zero mask probability at $t = 0$. Shared feature-wise tokenization and detokenization, together with a transformer encoder without positional encodings, make the denoiser equivariant to joint column and schema-entry permutations. Since the initialization and reverse-diffusion updates preserve this symmetry, the resulting distribution is column-permutation invariant.

C Experimental Details

C.1 Metrics

Shape. Shape measures the statistical similarity between column-marginal distributions of the real and synthetic data. It is based on the Kolmogorov-Smirnov (KS) test for numerical features and the

Total Variation (TV) distance for categorical features. Formally, let $\boldsymbol{X}^r$ and $\boldsymbol{X}^s$ denote the real and synthetic tables, respectively. The KS statistic corresponding to a numerical column j is defined as

$$\text{KS}(\boldsymbol{X}^r_{:,j}, \boldsymbol{X}^s_{:,j}) = \sup_u |\hat{F}^r_j(u) - \hat{F}^s_j(u)|,$$

where $\hat{F}^r_j$ and $\hat{F}^s_j$ are the empirical cumulative distribution functions (CDF) estimated from $\boldsymbol{X}^r_{:,j}$ and $\boldsymbol{X}^s_{:,j}$, respectively. For a categorical column j, the TV distance is defined as

$$\text{TV}(\boldsymbol{X}^r_{:,j}, \boldsymbol{X}^s_{:,j}) = \frac{1}{2} \sum_{v \in \mathcal{V}_j} |\hat{p}^r_j(v) - \hat{p}^s_j(v)|,$$

where $\hat{p}^r_j$ and $\hat{p}^s_j$ are the empirical probability mass functions (PMF) estimated from $\boldsymbol{X}^r_{:,j}$ and $\boldsymbol{X}^s_{:,j}$, respectively, and $\mathcal{V}_j$ is the vocabulary set of column j. The final Shape score is computed by aggregating the complements of these tests across all columns as follows

$$\text{Shape}_j = \begin{cases} 1 - \text{KS}(\boldsymbol{X}^r_{:,j}, \boldsymbol{X}^s_{:,j}) & \text{if } \tau_j = \text{num} \\ 1 - \text{TV}(\boldsymbol{X}^r_{:,j}, \boldsymbol{X}^s_{:,j}) & \text{if } \tau_j = \text{cat} \end{cases} \qquad \text{Shape}(\boldsymbol{X}^r, \boldsymbol{X}^s) = 100 \times \frac{1}{d} \sum_{j=1}^{d} \text{Shape}_j$$

Trend. Trend compares column-pair correlations between the real and synthetic data. For two numerical columns j and k, we compare their empirical Pearson correlations, ρ^r_{jk} and ρ^s_{jk}, estimated from the real and synthetic set, respectively, yielding the distance

$$D(\boldsymbol{X}^r_{:,jk}, \boldsymbol{X}^s_{:,jk}) = \frac{1}{2}|\rho^r_{jk} - \rho^s_{jk}|.$$

For categorical pairs, we compare their normalized contingency tables

$$D(\boldsymbol{X}^r_{:,jk}, \boldsymbol{X}^s_{:,jk}) = \frac{1}{2} \sum_{v \in \mathcal{V}_j, v' \in \mathcal{V}_k} |\hat{p}^r_{jk}(v, v') - \hat{p}^s_{jk}(v, v')|,$$

where $\hat{p}^r_{jk}(v, v')$ and $\hat{p}^s_{jk}(v, v')$ are the proportions of rows containing the category combination (v, v') in the real and synthetic sets, respectively. Mixed numerical-categorical pairs use the same contingency computation after discretizing the numerical column. The Trend score is obtained by normalizing and aggregating these distances across different pairs as follows

$$\text{Trend}_{jk} = 1 - D(\boldsymbol{X}^r_{:,jk}, \boldsymbol{X}^s_{:,jk}), \qquad \text{Trend} = 100 \times \frac{1}{|\mathcal{P}|} \sum_{(j,k) \in \mathcal{P}} \text{Trend}_{jk},$$

where $\mathcal{P}$ contains unordered column pairs with real data correlation strength above a pre-defined threshold. Both metrics are normalized to $[0, 100]$, with higher values indicating greater similarity.

α-Precision and β-Recall. These sample-level metrics were introduced by Alaa et al. (2022) and we use the implementation provided by Synthcity (Qian et al., 2023). Precision and recall measure fidelity and distributional coverage, respectively. To compute the overall scores, Alaa et al. (2022) introduce the notion of α-Precision, P_α, and β-Recall, R_β, defined respectively as

$$P_\alpha = P(\boldsymbol{x}^s \in \mathcal{S}^\alpha_r), \text{ for } \alpha \in [0, 1], \quad R_\beta = P(\boldsymbol{x}^r \in \mathcal{S}^\beta_s), \text{ for } \beta \in [0, 1],$$

where $\mathcal{S}^\alpha_r$ ($\mathcal{S}^\beta_s$) is a minimum-volume region containing probability mass α (β) under the real (synthetic) distribution. Intuitively, P_α corresponds to the fraction of synthetic samples that resemble the most typical fraction α of real samples, while R_β is the fraction of real samples covered by the most typical fraction β of synthetic samples.

In practice, these quantities are approximated by embedding the real and synthetic data into hyperspheres and computing the fraction of synthetic samples that fall within the α-quantile hypersphere (P_α) and a similar approach is used to approximate R_β. To aggregate the α-Precision and β-Recall values for different values of α and β into a single metric, we compute the mean absolute deviation from their optimal value, which represents a straight line with unity slope.

$$\Delta P_\alpha = \int_0^1 |P_\alpha - \alpha| d\alpha, \qquad \Delta R_\beta = \int_0^1 |R_\beta - \beta| d\beta,$$

which are in practice computed by evaluating the integrands at 30 equally-spaced grid points (Qian et al., 2023). The final scores normalize ΔP_α and ΔR_β into the range $[0, 100]$ with higher values indicating better distributional agreement

$$\alpha\text{-Precision} = (1 - 2\Delta P_\alpha) \times 100 \qquad \beta\text{-Recall} = (1 - 2\Delta R_\beta) \times 100.$$

Authenticity. Authenticity estimates potential memorization using the generator's training rows as the real reference by approximating the probability that the generator innovates a new sample rather than copies it from the training data (plus some noise). Alaa et al. (2022) propose to compute the Authenticity score as follows. For any synthetic sample $\boldsymbol{x}^s$, let $\boldsymbol{x}^{r*}(\boldsymbol{x}^s) = \arg\min_{\boldsymbol{x}^r \in \boldsymbol{X}^r} d(\boldsymbol{x}^s, \boldsymbol{x}^r)$ be its closest real sample according to the distance measure d. Further, let $D(\boldsymbol{x}) = \min_{\boldsymbol{x}^r \in \boldsymbol{X}^r \setminus \{\boldsymbol{x}\}} d(\boldsymbol{x}, \boldsymbol{x}^r)$ be the smallest distance between a real sample $\boldsymbol{x}$ and any other real sample. Then, the Authenticity score is estimated as

$$\text{Authenticity} = 100 \times \frac{1}{|\boldsymbol{X}^s|} \sum_{\boldsymbol{x}^s \in \boldsymbol{X}^s} \mathbf{1}[d(\boldsymbol{x}^s, \boldsymbol{x}^{r*}(\boldsymbol{x}^s)) > D(\boldsymbol{x}^{r*}(\boldsymbol{x}^s))],$$

which effectively computes the fraction of synthetic rows whose distance to the nearest training record exceeds that record's distance to its nearest other training record.

Machine Learning Efficacy (MLE). We evaluate the predictive utility of synthetic data by training an XGBoost classifier or regressor on the synthetic dataset and evaluating it on the held-out test dataset. Following TabSyn (Zhang et al., 2024), we divide the synthetic dataset into training and validation subsets with an 8:1 ratio and select hyperparameters using validation performance, and the best model is tested on the test dataset mimicking real-world development lifecycles. For classification datasets, we compute the Area Under the Receiver Operating Characteristic Curve (AUC). For regression datasets, we compute the Normalized Root Mean Squared Error $\text{NRMSE} = \frac{\text{RMSE}}{y_{\max} - y_{\min}}$, where RMSE is computed between predicted and ground-truth targets in the test set and $y_{\max}$ and $y_{\min}$ are the largest and smallest target values, respectively. For all datasets, to obtain a score in $[0, 100]$, with higher values indicating better utility, we report

$$\text{MLE} = \begin{cases} 100 \times \text{AUC}, & \text{for classification datasets,} \\ 100 \times \max(0, 1 - \text{NRMSE}), & \text{for regression datasets,} \end{cases}$$

where the $\max$ operator avoids getting a negative score when the RMSE is higher than the target range.

Distance to closest record (DCR). DCR score evaluates whether synthetic records are systematically closer to the generator's training data than to holdout data. It is based on Gower's distance, which is a distance measure between mixed-type objects. For two mixed-type rows $\boldsymbol{x}$ and $\boldsymbol{x}'$, this distance is defined as

$$D(\boldsymbol{x}, \boldsymbol{x}') = \sum_{j \in \mathcal{J}_{\text{num}}} \frac{|x_j - x'_j|}{R_j} + \sum_{j \in \mathcal{J}_{\text{cat}}} \mathbf{1}[x_j \neq x'_j],$$

where $R_j = \max_{\boldsymbol{x} \in \boldsymbol{X}} x_j - \min_{\boldsymbol{x} \in \boldsymbol{X}} x_j$ is the range of numerical feature j. For each synthetic record $\boldsymbol{x}^s \in \boldsymbol{X}_{\text{syn}}$, we compute its distance to the closest neighbor from the equally-sized train set and from the holdout test set:

$$d_T(\boldsymbol{x}^s) = \min_{\boldsymbol{x}^t \in \boldsymbol{X}_{\text{train}}} D(\boldsymbol{x}^s, \boldsymbol{x}^t), \qquad d_H(\boldsymbol{x}^s) = \min_{\boldsymbol{x}^h \in \boldsymbol{X}_{\text{holdout}}} D(\boldsymbol{x}^s, \boldsymbol{x}^h).$$

The raw score is the fraction of synthetic rows that are closer to the training set than to the holdout set

$$q = \frac{1}{N} \sum_{i=1}^{N} \mathbf{1}[d_T(\boldsymbol{x}_i^s) < d_H(\boldsymbol{x}_i^s)]$$

Values of $q \leq 0.5$ indicate the absence of excessive proximity to the train set and are preferred. This is reflected in the DCR score normalization

$$\text{DCR} = 100 \times \min(1, 2(1 - q)).$$

C.2 Evaluation datasets

We evaluate on the six datasets used by TabSyn (Zhang et al., 2024), together with Diabetes, as in TabDiff (Shi et al., 2025a). These datasets cover diverse domains and prediction tasks:

Table 2: Evaluation dataset statistics. Numerical and categorical column counts include the target. Max Cat denotes the cardinality of the largest categorical vocabulary; Binary denotes binary classification.

Dataset	# Train Rows	# Test Rows	# Cols	# Num	# Cat	# Max Cat	Task
Adult	24,421	24,421	15	6	9	42	Binary
Default	15,000	15,000	24	14	10	11	Binary
Magic	9,510	9,510	11	10	1	2	Binary
Shoppers	6,165	6,165	18	10	8	20	Binary
Diabetes	50,883	50,883	37	8	29	66	Multiclass
Beijing	20,878	20,879	8	7	1	4	Regression
News	19,822	19,822	47	45	2	7	Regression

- Adult[1] (demographics): Census records containing demographic and employment attributes, with a binary classification task predicting whether annual income exceeds USD 50,000.
- Default[2] (finance): Credit-card client records containing demographic, credit, and repayment information, with a binary classification task predicting payment default.
- Magic[3] (astroparticle physics): Simulated telescope observations, with a binary classification task distinguishing gamma-ray events from hadronic background.
- Shoppers[4] (e-commerce): Online browsing-session records, with a binary classification task predicting whether a session results in a purchase.
- Diabetes[5] (healthcare): Demographic and clinical records of hospital encounters involving patients with diabetes. We retain the original three readmission classes: less than 30 days, more than 30 days, and no readmission, yielding a multiclass classification task.
- Beijing[6] (environmental monitoring): Air-quality and meteorological measurements, with a regression task predicting PM 2.5 concentrations.
- News[7] (online media): Characteristics of online news articles, with a regression task predicting the number of social-media shares.

Table 2 summarizes the dataset sizes, feature composition, categorical cardinalities, and prediction tasks.

C.3 Pre-training datasets

Table 3: Pretraining dataset summary across the 337 datasets.

Quantity	Mean	Median	Min	Max	Sum
# Rows	28,585	10,999	10	100,000	9,633,139
# Cols	11.1	10	2	33	3,734
# Num	7.3	6	0	28	2,458
# Cat	3.8	2	0	23	1,276
Total datasets: 337					

[1] https://archive.ics.uci.edu/dataset/2/adult
[2] https://archive.ics.uci.edu/dataset/350/default+of+credit+card+clients
[3] https://archive.ics.uci.edu/dataset/159/magic+gamma+telescope
[4] https://archive.ics.uci.edu/dataset/468/online+shoppers+purchasing+intention+dataset
[5] https://archive.ics.uci.edu/dataset/296/diabetes+130-us+hospitals+for+years+1999-2008
[6] https://archive.ics.uci.edu/dataset/381/beijing+pm2+5+data
[7] https://archive.ics.uci.edu/dataset/332/online+news+popularity

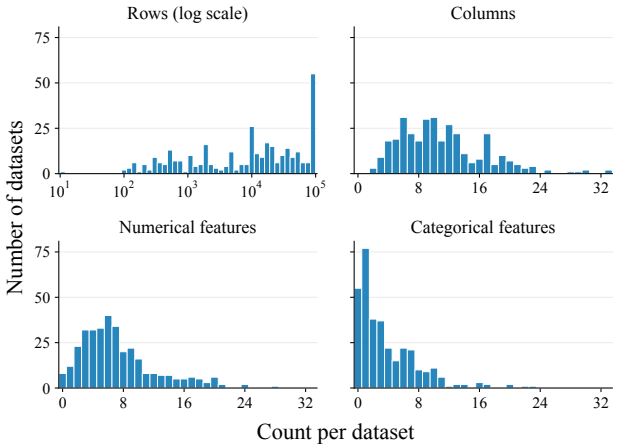

Figure 4: Distribution of row counts and total, numerical, and categorical column counts across the 337 pre-training datasets. Each dataset contributes one observation, and column counts reflect retained features after preprocessing. Row counts use 50 logarithmically spaced bins; column counts use one bin per integer.

We construct our pre-training corpus from the datasets curated by Yan et al. (2024), drawn from TabPertNet, now known as OpenTabs (Ye et al., 2024). This source collection contains heterogeneous tables from OpenML, UCI, Kaggle, and Data.gov, with numerical and categorical features. The curated subset covers binary classification and regression tasks and excludes datasets with uninformative column names or categorical codes that cannot be mapped to meaningful values. The same source collection also underlies the pre-training data used by CTSYN (Lin et al., 2025).

To identify potential overlap with our evaluation datasets, we automatically compare dataset names, column names, marginal distributions, and categorical values between each evaluation dataset and every dataset in the pre-training pool. We then manually inspect the flagged datasets and remove those identified as copies or subsets of the evaluation datasets. This procedure removes 10 of the initial 347 datasets, leaving 337 datasets for pre-training. Table 3 and Figure 4 summarize the resulting corpus.

C.4 Implementation Details

Data pre-processing. We impute missing numerical values with the corresponding training column mean and represent missing categorical values with a dedicated category. To reduce differences in scale and marginal distributions across numerical features, we apply a quantile transformation. Quantile transformations are fitted exclusively on the training data; in few-shot settings, only the N available training samples are used for fitting. After sampling, we apply the corresponding inverse transformations to map generated numerical values back to their original scale.

Baselines. We train and evaluate all baselines under the same experimental protocol as our model, using identical data splits, training subsets, and evaluation metrics. For each baseline, we use its official implementation with the default hyperparameters, without additional tuning. Since our evaluation datasets follow established benchmarks used in prior work, we retain the configurations provided by the original authors. For two-stage models, we count one epoch as a complete pass over the data by the autoencoder and the latent diffusion model.

CDMD hyperparameters. We optimize our model using AdamW with a learning rate of 10^{-4} for 2,000 epochs, using a batch size of 4096. We apply a linear learning-rate warmup over the first 5% of training steps, and subsequently multiply the learning rate by 0.9 whenever the training loss does not improve for 50 consecutive epochs. During few-shot fine-tuning, we freeze the transformer encoder. We generate synthetic samples using 50 sampling steps on a linear time grid. In the

Table 4: In-domain cross-dataset generation results for the three evaluation axes

Fidelity

	Method \ Dataset	Adult	Default	Magic	Shoppers	Diabetes	Beijing	News	Average
	TRAIN SET	86.52 ±0.00	87.06 ±0.00	86.61 ±0.00	86.37 ±0.00	86.84 ±0.00	87.21 ±0.00	85.07 ±0.00	86.53
Single-dataset	SMOTE	82.35 ±0.10	84.31 ±0.16	84.49 ±0.26	82.26 ±0.19	81.05 ±0.08	**84.75** ±0.04	81.39 ±0.80	82.94
	CTGAN	67.62 ±2.70	64.91 ±1.55	66.43 ±1.12	67.75 ±1.22	64.35 ±4.79	75.59 ±1.02	66.04 ±0.66	67.53
	TVAE	76.38 ±1.56	73.32 ±1.12	76.98 ±0.39	57.64 ±0.73	60.59 ±1.65	76.11 ±1.65	73.12 ±0.51	70.59
	GREAT	74.05 ±0.14	77.30 ±0.32	74.07 ±0.40	75.88 ±0.29	73.92 ±0.11	80.01 ±0.08	–	75.87
	TABDDPM	84.71 ±0.10	83.12 ±0.34	84.69 ±0.17	80.42 ±0.14	32.44 ±0.00	83.09 ±0.07	18.83 ±0.02	66.76
	TABSYN	85.81 ±0.02	85.38 ±0.07	85.58 ±0.10	84.54 ±0.39	80.28 ±0.00	83.23 ±0.10	80.14 ±0.13	83.57
	TABDIFF	85.53 ±0.02	85.68 ±0.03	85.53 ±0.09	84.18 ±0.37	84.25 ±0.04	83.44 ±0.09	73.66 ±0.08	83.18
Cross-dataset	TABEBM	38.07 ±0.08	50.41 ±0.20	78.75 ±0.04	51.66 ±0.19	29.86 ±0.28	68.01 ±0.15	53.77 ±11.75	52.93
	CTSYN	58.68 ±0.10	72.77 ±0.11	66.30 ±0.02	65.52 ±0.14	43.55 ±0.02	66.93 ±0.10	70.61 ±0.07	63.48
	CDMD (OURS)	**85.97** ±0.02	**86.08** ±0.07	**85.66** ±0.16	**85.39** ±0.13	**85.60** ±0.07	83.36 ±0.02	**82.47** ±0.06	**84.93**

Utility

	Method \ Dataset	Adult	Default	Magic	Shoppers	Diabetes	Beijing	News	Average
	TRAIN SET	92.75 ±0.00	77.98 ±0.00	92.78 ±0.00	92.76 ±0.00	67.97 ±0.00	89.09 ±0.00	89.86 ±0.00	86.17
Single-dataset	SMOTE	90.00 ±0.11	74.35 ±0.25	**92.32** ±0.04	91.20 ±0.24	63.78 ±0.05	88.70 ±0.06	**89.92** ±0.04	84.32
	CTGAN	88.49 ±0.20	71.96 ±0.73	82.68 ±2.44	81.64 ±3.94	55.57 ±0.89	86.97 ±0.36	89.24 ±0.13	79.51
	TVAE	88.57 ±0.22	74.99 ±0.33	83.53 ±2.39	88.20 ±0.38	60.46 ±1.16	87.11 ±0.27	87.78 ±0.17	81.52
	GREAT	**91.22** ±0.29	75.79 ±0.79	90.01 ±0.14	91.03 ±0.43	**66.73** ±0.34	88.28 ±0.20	–	83.84
	TABDDPM	90.81 ±0.11	76.58 ±0.46	91.97 ±0.06	90.37 ±0.17	50.51 ±0.36	88.92 ±0.56	58.35 ±11.80	78.21
	TABSYN	91.05 ±0.14	**77.34** ±0.23	92.11 ±0.10	91.15 ±0.53	64.38 ±0.17	**89.13** ±0.24	89.60 ±0.30	84.97
	TABDIFF	91.04 ±0.18	77.09 ±0.44	91.82 ±0.08	**91.68** ±0.22	66.50 ±0.31	88.86 ±0.53	89.85 ±0.27	**85.26**
Cross-dataset	TABEBM	86.73 ±1.24	64.40 ±1.80	89.75 ±0.37	88.33 ±1.22	57.07 ±0.55	86.61 ±0.36	81.56 ±0.58	79.21
	CTSYN	76.63 ±1.14	60.82 ±1.48	78.63 ±0.71	61.82 ±3.46	52.52 ±0.06	85.20 ±0.18	88.70 ±0.19	72.05
	CDMD (OURS)	91.21 ±0.07	77.27 ±0.21	92.18 ±0.09	91.58 ±0.15	66.58 ±0.04	88.43 ±0.52	89.54 ±0.02	85.26

Privacy

	Method \ Dataset	Adult	Default	Magic	Shoppers	Diabetes	Beijing	News	Average
	TRAIN SET	0.09 ±0.00	0.11 ±0.00	0.60 ±0.00	1.14 ±0.00	0.00 ±0.00	0.01 ±0.00	0.00 ±0.00	0.28
Single-dataset	SMOTE	23.55 ±0.18	23.71 ±0.17	16.99 ±0.27	16.93 ±0.65	12.58 ±0.07	26.42 ±0.15	14.35 ±0.16	19.22
	CTGAN	78.91 ±0.28	79.75 ±0.63	87.67 ±0.14	80.31 ±0.26	86.71 ±1.01	85.97 ±0.33	85.51 ±0.45	83.55
	TVAE	77.46 ±0.51	81.76 ±1.16	81.41 ±1.28	**87.05** ±2.03	82.35 ±0.51	84.58 ±0.87	81.29 ±0.28	82.27
	GREAT	73.20 ±0.16	75.13 ±0.45	76.84 ±0.51	76.04 ±0.48	74.21 ±0.22	73.25 ±0.15	–	74.78
	TABDDPM	74.51 ±0.33	74.68 ±0.07	72.88 ±0.35	65.11 ±0.32	**98.65** ±0.17	81.28 ±0.26	**100.00** ±0.00	81.02
	TABSYN	73.97 ±0.10	74.28 ±0.28	74.27 ±0.30	74.21 ±0.88	78.47 ±0.19	80.61 ±0.13	77.41 ±0.25	76.18
	TABDIFF	66.89 ±0.19	66.91 ±0.32	73.61 ±0.31	74.43 ±0.21	67.64 ±0.35	80.58 ±0.31	74.78 ±0.13	72.12
Cross-dataset	TABEBM	83.85 ±0.37	**89.15** ±0.66	40.59 ±0.21	75.87 ±0.70	93.43 ±0.04	82.30 ±0.30	53.62 ±0.31	74.12
	CTSYN	**84.48** ±0.07	81.56 ±0.35	**89.29** ±0.72	81.27 ±1.08	92.35 ±0.05	**96.46** ±0.07	79.49 ±0.40	**86.41**
	CDMD (OURS)	74.12 ±0.50	74.00 ±0.79	74.13 ±0.70	72.40 ±0.24	75.34 ±0.15	80.45 ±0.19	75.35 ±0.19	75.11

few-epoch adaptation experiment, we do not apply the learning-rate warmup stage and train for the specified number of epochs.

D DETAILED RESULTS

In this section, we provide detailed evaluation results for our experiments presented in Section 4.

D.1 CROSS-DATASET GENERATION

For the in-domain cross-dataset generation experiments (Section 4.1), we present per-axis evaluation results in Table 4. All four fidelity metrics are in Table 5 and the two privacy metrics are in Table 6. Since Utility has only one metric, the metric score is the same as the aggregate score presented in Table 4.

D.2 FEW-SHOT GENERATION

For the few-shot generation experiments (Section 4.2), we complement the results presented in the main paper with the generation quality score in Figure 5.

Table 5: In-domain cross-dataset generation results for the four individual metrics of Fidelity.

Shape

	Method \ Dataset	Adult	Default	Magic	Shoppers	Diabetes	Beijing	News	Average
	TRAIN SET	99.38 ±0.00	99.23 ±0.00	99.08 ±0.00	98.84 ±0.00	99.73 ±0.00	99.32 ±0.00	99.20 ±0.00	99.25
Single-dataset	SMOTE	98.22 ±0.03	97.99 ±0.10	98.47 ±0.08	97.48 ±0.04	98.48 ±0.01	**99.18** ±0.10	96.28 ±0.04	98.02
	CTGAN	85.73 ±2.07	85.51 ±0.64	89.27 ±1.10	81.14 ±1.38	89.91 ±0.50	92.52 ±1.50	79.45 ±0.96	86.22
	TVAE	89.32 ±1.38	89.85 ±0.33	92.45 ±0.75	77.50 ±0.36	91.44 ±0.72	93.82 ±1.92	82.50 ±0.29	88.13
	GREAT	91.86 ±0.08	85.88 ±0.27	83.92 ±0.06	85.74 ±0.10	93.83 ±0.02	89.57 ±0.12	–	88.47
	TABDDPM	98.71 ±0.03	95.71 ±0.11	98.24 ±0.16	95.16 ±0.07	70.27 ±0.01	98.88 ±0.12	14.05 ±0.07	81.57
	TABSYN	99.09 ±0.03	98.59 ±0.03	98.56 ±0.08	97.62 ±0.16	97.92 ±0.02	98.94 ±0.04	95.91 ±0.02	98.09
	TABDIFF	98.98 ±0.10	98.39 ±0.10	**98.76** ±0.05	97.94 ±0.10	99.05 ±0.03	99.00 ±0.14	96.77 ±0.01	98.41
Cross-dataset	TABEBM	70.86 ±0.03	75.57 ±0.37	96.06 ±0.00	72.98 ±0.04	69.21 ±0.77	79.21 ±0.44	82.74 ±0.79	78.09
	CTSYN	89.77 ±0.03	91.30 ±0.05	84.63 ±0.07	88.16 ±0.03	88.73 ±0.02	87.76 ±0.02	89.27 ±0.07	88.52
	CDMD (OURS)	**99.18** ±0.10	**98.83** ±0.07	98.70 ±0.20	**98.47** ±0.01	**99.54** ±0.02	99.07 ±0.07	**98.48** ±0.02	**98.90**

Trend

	Method \ Dataset	Adult	Default	Magic	Shoppers	Diabetes	Beijing	News	Average
	TRAIN SET	98.65 ±0.00	98.95 ±0.00	97.60 ±0.00	98.47 ±0.00	99.30 ±0.00	99.96 ±0.00	91.90 ±0.00	97.83
Single-dataset	SMOTE	97.19 ±0.01	97.19 ±0.08	95.97 ±0.84	97.17 ±0.11	97.19 ±0.10	99.37 ±0.05	**92.29** ±3.38	96.63
	CTGAN	80.80 ±2.16	82.67 ±1.23	86.10 ±0.55	80.81 ±0.10	80.95 ±1.91	98.03 ±0.64	83.31 ±0.82	84.67
	TVAE	89.86 ±0.56	93.86 ±0.17	91.26 ±1.65	89.97 ±0.89	93.52 ±1.77	95.56 ±0.54	90.13 ±0.49	92.02
	GREAT	88.34 ±0.20	90.54 ±0.30	88.59 ±1.46	91.52 ±1.06	87.88 ±0.19	98.13 ±0.01	–	90.83
	TABDDPM	97.25 ±0.08	95.13 ±0.13	96.41 ±0.30	87.10 ±0.07	48.35 ±0.01	99.36 ±0.05	61.28 ±0.09	83.55
	TABSYN	97.83 ±0.03	98.19 ±0.08	97.10 ±0.16	96.86 ±0.24	96.01 ±0.12	99.92 ±0.02	91.48 ±0.32	96.77
	TABDIFF	97.80 ±0.12	98.29 ±0.06	97.04 ±0.36	96.48 ±0.27	98.49 ±0.08	**99.94** ±0.00	84.19 ±0.14	96.03
Cross-dataset	TABEBM	55.93 ±0.10	73.00 ±0.03	91.71 ±0.13	73.21 ±0.19	48.71 ±0.02	99.22 ±0.03	88.02 ±3.98	75.68
	CTSYN	73.59 ±0.12	91.02 ±0.06	82.08 ±0.01	82.05 ±0.19	73.98 ±0.06	91.21 ±0.01	87.58 ±0.22	83.07
	CDMD (OURS)	**98.18** ±0.06	**98.58** ±0.05	**97.32** ±0.18	**97.65** ±0.22	**99.10** ±0.09	99.86 ±0.05	91.52 ±0.09	**97.46**

Precision

	Method \ Dataset	Adult	Default	Magic	Shoppers	Diabetes	Beijing	News	Average
	TRAIN SET	99.52 ±0.00	99.69 ±0.00	99.54 ±0.00	98.77 ±0.00	99.37 ±0.00	99.42 ±0.00	99.47 ±0.00	99.40
Single-dataset	SMOTE	92.63 ±0.28	97.19 ±0.46	97.80 ±0.12	92.60 ±0.43	87.01 ±0.21	98.99 ±0.22	87.44 ±0.58	93.38
	CTGAN	77.25 ±5.40	73.24 ±3.44	81.36 ±3.05	83.43 ±3.92	73.77 ±17.87	87.49 ±2.17	91.67 ±1.51	81.17
	TVAE	93.25 ±2.10	86.23 ±2.81	95.36 ±1.69	49.53 ±1.22	39.19 ±1.64	90.33 ±2.83	93.85 ±1.09	78.25
	GREAT	66.68 ±0.37	89.89 ±0.34	84.68 ±0.16	80.75 ±0.92	59.51 ±0.21	96.19 ±0.13	–	79.62
	TABDDPM	95.98 ±0.08	96.62 ±0.37	95.98 ±0.32	91.27 ±0.22	11.10 ±0.03	99.20 ±0.39	0.00 ±0.00	70.02
	TABSYN	99.35 ±0.14	98.59 ±0.07	98.61 ±0.47	97.90 ±0.40	92.27 ±0.20	98.67 ±0.16	96.95 ±0.14	97.48
	TABDIFF	98.93 ±0.31	98.56 ±0.05	**99.49** ±0.07	97.29 ±0.97	93.69 ±0.26	99.39 ±0.12	86.73 ±0.12	96.30
Cross-dataset	TABEBM	24.17 ±0.27	46.68 ±0.46	92.89 ±0.08	47.06 ±0.44	1.51 ±0.35	78.75 ±0.86	38.45 ±34.35	47.07
	CTSYN	53.78 ±0.25	89.97 ±0.44	87.31 ±0.12	66.48 ±0.30	8.32 ±0.01	83.98 ±0.27	77.85 ±0.06	66.81
	CDMD (OURS)	**99.59** ±0.06	**98.77** ±0.06	99.22 ±0.43	**98.65** ±0.24	**99.64** ±0.07	**99.52** ±0.13	**98.50** ±0.29	**99.13**

Recall

	Method \ Dataset	Adult	Default	Magic	Shoppers	Diabetes	Beijing	News	Average
	TRAIN SET	48.55 ±0.00	50.39 ±0.00	50.20 ±0.00	49.40 ±0.00	48.94 ±0.00	50.12 ±0.00	49.71 ±0.00	49.62
Single-dataset	SMOTE	41.35 ±0.30	44.87 ±0.04	45.70 ±0.25	41.76 ±0.24	41.51 ±0.04	**41.43** ±0.29	**49.56** ±0.24	43.74
	CTGAN	26.70 ±1.75	18.23 ±1.01	9.01 ±0.73	25.62 ±0.83	12.76 ±2.10	24.32 ±1.11	9.75 ±0.06	18.06
	TVAE	33.07 ±2.48	23.35 ±1.42	28.86 ±1.88	13.56 ±1.31	18.24 ±2.74	24.74 ±1.67	26.00 ±0.80	23.97
	GREAT	**49.31** ±0.08	42.88 ±0.55	39.10 ±0.07	45.50 ±0.21	**54.44** ±0.08	36.13 ±0.11	–	**44.56**
	TABDDPM	46.90 ±0.23	45.04 ±0.97	**48.13** ±0.30	**48.16** ±0.64	0.04 ±0.01	34.93 ±0.23	0.00 ±0.00	31.88
	TABSYN	46.98 ±0.10	46.14 ±0.22	48.06 ±0.29	45.78 ±0.83	34.93 ±0.15	35.40 ±0.27	36.20 ±0.15	41.93
	TABDIFF	46.39 ±0.24	47.50 ±0.30	46.83 ±0.67	45.03 ±0.22	45.79 ±0.03	35.44 ±0.35	26.94 ±0.24	41.99
Cross-dataset	TABEBM	1.32 ±0.02	6.41 ±0.19	34.35 ±0.05	13.41 ±0.07	0.00 ±0.00	14.87 ±0.25	5.87 ±7.93	10.89
	CTSYN	17.57 ±0.18	18.80 ±0.08	11.19 ±0.12	25.36 ±0.28	3.18 ±0.01	4.78 ±0.12	27.76 ±0.23	15.52
	CDMD (OURS)	46.93 ±0.14	**48.15** ±0.33	47.41 ±0.23	46.77 ±0.04	44.13 ±0.14	34.98 ±0.14	41.36 ±0.03	44.25

D.3 FEW-EPOCH GENERATION

For the few-epoch generation experiments (Section 4.2), we plot the generation quality score against the number of training epochs in Figure 6.

Table 6: In-domain cross-dataset generation results for Privacy.

DCR

	Method \ Dataset	Adult	Default	Magic	Shoppers	Diabetes	Beijing	News	Average
	TRAIN SET	0.19 ±0.00	0.21 ±0.00	1.20 ±0.00	2.27 ±0.00	0.01 ±0.00	0.03 ±0.00	0.00 ±0.00	0.56
Single-dataset	SMOTE	18.23 ±0.12	16.94 ±0.13	8.76 ±0.29	7.66 ±0.76	1.11 ±0.09	23.26 ±0.27	3.52 ±0.18	11.35
	CTGAN	98.31 ±0.20	97.46 ±1.01	98.79 ±0.44	97.81 ±0.66	**100.00** ±0.00	99.94 ±0.09	96.98 ±0.78	98.47
	TVAE	98.58 ±0.14	**98.77** ±0.90	99.08 ±1.06	97.08 ±2.78	98.71 ±0.68	98.09 ±0.98	96.27 ±0.80	98.09
	GREAT	95.15 ±0.23	98.12 ±0.72	97.26 ±0.30	**99.75** ±0.34	97.97 ±0.43	87.16 ±0.31	–	95.90
	TABDDPM	97.76 ±0.32	98.22 ±0.23	94.31 ±0.17	83.42 ±0.62	98.77 ±0.34	99.31 ±0.45	**100.00** ±0.00	95.97
	TABSYN	96.75 ±0.25	97.97 ±0.38	96.80 ±0.29	97.33 ±1.58	99.72 ±0.39	98.43 ±0.41	98.22 ±0.34	97.89
	TABDIFF	85.55 ±0.20	85.69 ±0.57	95.46 ±0.21	97.11 ±0.20	86.15 ±0.51	98.47 ±0.41	91.89 ±0.11	91.47
Cross-dataset	TABEBM	82.08 ±0.66	95.01 ±0.29	40.14 ±0.23	80.30 ±0.61	86.89 ±0.10	87.89 ±0.54	48.93 ±0.61	74.46
	CTSYN	**100.00** ±0.00	98.51 ±0.57	**100.00** ±0.00	95.81 ±1.78	**100.00** ±0.00	**100.00** ±0.00	96.72 ±0.88	**98.72**
	CDMD (OURS)	97.25 ±0.64	97.79 ±1.04	96.52 ±1.42	95.50 ±0.48	98.53 ±0.27	98.13 ±0.30	97.04 ±0.17	97.25

Authenticity

	Method \ Dataset	Adult	Default	Magic	Shoppers	Diabetes	Beijing	News	Average
	TRAIN SET	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00	0.00
Single-dataset	SMOTE	28.87 ±0.40	30.49 ±0.23	25.21 ±0.36	26.20 ±0.56	24.04 ±0.11	29.58 ±0.14	25.18 ±0.20	27.08
	CTGAN	59.51 ±0.46	62.05 ±0.49	76.55 ±0.26	62.81 ±0.20	73.42 ±2.02	72.01 ±0.64	74.04 ±0.60	68.63
	TVAE	56.33 ±0.91	64.75 ±1.49	63.73 ±1.58	**77.02** ±1.29	65.99 ±1.70	71.07 ±2.27	66.32 ±1.19	66.46
	GREAT	51.24 ±0.12	52.14 ±0.39	56.42 ±0.80	52.33 ±0.63	50.45 ±0.23	59.34 ±0.22	–	53.65
	TABDDPM	51.26 ±0.38	51.14 ±0.28	51.45 ±0.53	46.81 ±0.50	98.53 ±0.15	63.25 ±0.08	**100.00** ±0.00	66.06
	TABSYN	51.20 ±0.34	50.59 ±0.20	51.74 ±0.62	51.08 ±0.24	57.23 ±0.09	62.78 ±0.26	56.61 ±0.17	54.46
	TABDIFF	48.22 ±0.27	48.14 ±0.18	51.76 ±0.67	51.75 ±0.61	49.12 ±0.22	62.69 ±0.20	57.68 ±0.14	52.77
Cross-dataset	TABEBM	**85.62** ±1.17	**83.29** ±1.06	41.04 ±0.19	71.44 ±1.07	**99.97** ±0.01	76.71 ±0.32	58.31 ±0.10	73.77
	CTSYN	68.95 ±0.13	64.60 ±1.03	**78.58** ±1.44	66.73 ±0.51	84.70 ±0.11	**92.92** ±0.13	62.26 ±0.15	**74.11**
	CDMD (OURS)	51.00 ±0.39	50.21 ±0.55	51.75 ±0.55	49.30 ±0.09	52.14 ±0.20	62.77 ±0.30	53.66 ±0.25	52.97

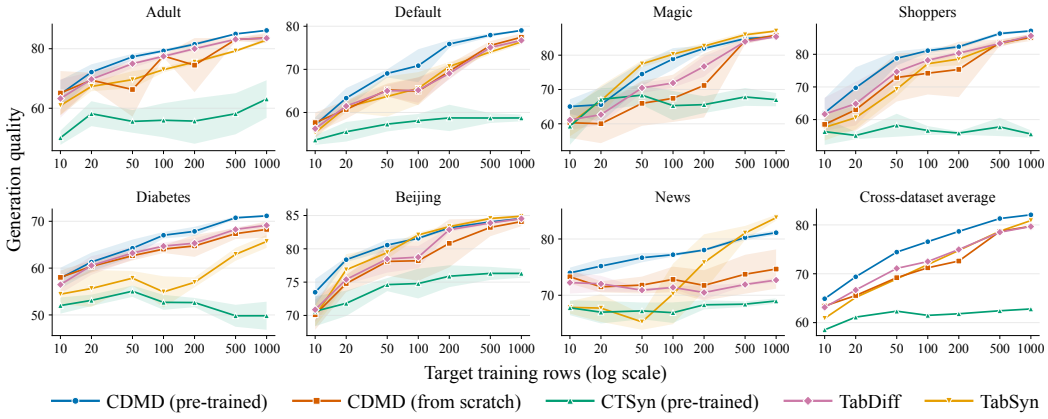

Figure 5: Few-shot performance across seven datasets and their cross-dataset average. Curves show mean performance, and shaded regions indicate one standard deviation across seeds.

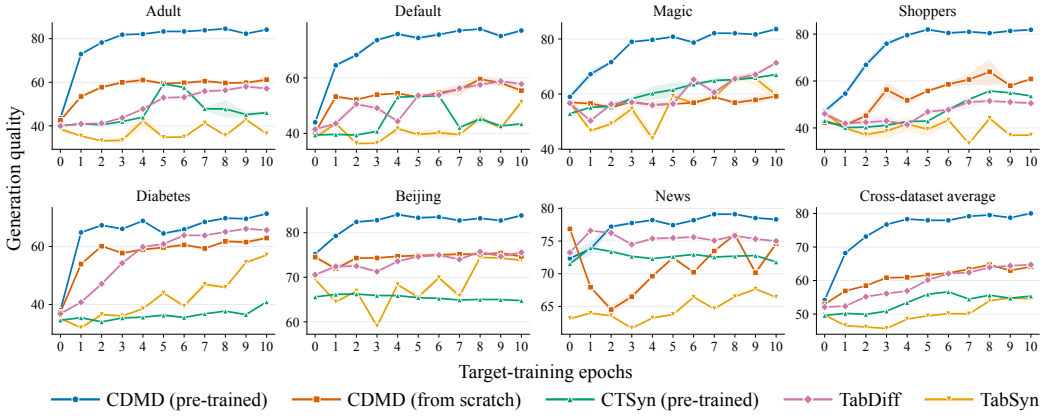

Figure 6: Few-epoch performance across seven datasets and their cross-dataset average. Curves show mean performance, and shaded regions indicate one standard deviation across seeds.