

# TABGEN: Training Tabular Diffusion Models with a Simple and Effective Continuous Representation

Anonymous Authors<sup>1</sup>

## Abstract

Diffusion models for tabular data generation face the conundrum of modeling under a separate versus a unified data representation. The former encounters the challenge of jointly modeling all multi-modal distributions of tabular data in one model. While the latter alleviates this by learning a single representation for all features, it currently leverages sparse suboptimal encoding heuristics and necessitates additional computation costs. In this work, we address the latter by presenting TABGEN, a tabular diffusion architecture trained with a unified continuous representation, the CATCONVERTER. To motivate the design, we provide geometric insights into how the data manifold affects diffusion models. The key attributes of the CATCONVERTER are composed of its dense representation, flexibility to provide ample separability for nominal features, and ability to preserve intrinsic relationships. Ultimately, TABGEN provides a simple yet effective approach for training tabular diffusion models under a continuous data manifold. Our results showcase that TABGEN achieves superior performance across a broad suite of evaluations. It is the first to synthesize tabular data that exceeds the downstream quality of the original datasets while preserving privacy and remaining computationally efficient.

## 1. Introduction

Tabular data are ubiquitous in data ecosystems of many sectors such as healthcare, finance, and insurance (Clore et al., 2014; Moro et al., 2012; Datta, 2020). These industries utilize tabular data generation for many practical purposes, including data augmentation (Jolicoeur-Martineau et al., 2024), privacy-preserving machine learning (Xu et al.,

2021), and handling sparse, imbalanced datasets (Onishi & Meguro, 2023; Sauber-Cole & Khoshgoftaar, 2022). Unlike homogeneous data modalities such as images or text, one notable characteristic inherent to tabular data is feature heterogeneity (Liu et al., 2023). Tabular data often contain mixed feature types, ranging from (dense) continuous features to (sparse) categorical features.

Recently, the best-performing tabular generative models have been based on diffusion models (Ho et al., 2020b; Song & Ermon, 2020). For instance, Lee et al. (2023); Kotelnikov et al. (2023); Shi et al. (2024b) leverage continuous (Ho et al., 2020b; Song & Ermon, 2020) and discrete diffusion (Hooeboom et al., 2021; Shi et al., 2024a) to generate tabular data. However, these methods are designed to jointly optimize a multimodal continuous-discrete data representation that complicates the training process. On the other hand, while solutions that optimize a unified continuous representation circumvent this by learning a single representation for all features, they rely on encoding heuristics such as one-hot (Kim et al., 2022) or learned latent embeddings using a  $\beta$ -VAE (Zhang et al., 2023). Unfortunately, one-hot encoding for categorical variables leads to sparse representations in high dimensions, where generative models are susceptible to underfitting (Krishnan et al., 2017; Poslavskaya & Korolev, 2023). Furthermore, using a latent embedding space (Zhang et al., 2023) requires training the additional VAE embedding model (Higgins et al., 2017; Kingma & Welling, 2013) that requires extra computation while being dependent on the quality of the latent space.

Since diffusion-based generative models rely on continuous transformations of denoising score-matching, or invertible mappings between data and latent spaces, designing an effective data representation is key for high-performing diffusion models (Bengio et al., 2014). In this work, we propose TABGEN, a simple and effective continuous representation for training tabular diffusion models. We first introduce geometric insights to understand the properties of a continuous data manifold for tabular diffusion models. Then, we craft an abstract and useful representation, the CATCONVERTER, that enables diffusion models like DDPM (Ho et al., 2020a) and Flow Matching (Lipman et al., 2022) to capture the posterior distribution efficaciously. First, CATCONVERTER

<sup>1</sup>Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

maintains a dense two-dimensional representation, circumventing the curse of dimensionality that one-hot encoding faces. Next, CATCONVERTER is also well-suited to model the cardinality of nominal features, ensuring ample separability to distinguish between different categories. Lastly, coupling the cyclical nature of CATCONVERTER with the separability it offers enables it to facilitate both ordinal and periodic features. These attributes present desirable characteristics that make it easier for diffusion models to extract meaningful information from a unified continuous tabular data representation. We conduct comprehensive experiments against state-of-the-art diffusion baselines across various datasets and benchmarks. The results showcase that TABGEN consistently outperforms these baselines in quality, privacy-preservation, and computation cost indicating our superior capabilities to generate tabular data.

## 2. Related Work

The latest tabular diffusion models have made considerable progress compared to previous generative models such as VAEs (Xu et al., 2019) and GANs (Xu et al., 2019). This included STaSy (Kim et al., 2022), which employed a score-matching diffusion model paired with self-paced learning and fine-tuning to stabilize the training process, and CoDi (Lee et al., 2023), which used separate diffusion schemes for categorical and numerical data along with interconditioning and contrastive learning to improve synergy among features. TabDDPM (Kotelnikov et al., 2023) presented a similar diffusion scheme compared to CoDi and showed that the simple concatenation of categorical and numerical data before and after denoising led to improvements in performance. TabSYN (Zhang et al., 2023) is a latent diffusion model that transformed features into a unified embedding via a  $\beta$ -VAE (Kingma & Welling, 2013) before applying EDM diffusion (Karras et al., 2022) to generate synthetic data. The latest model, TabDiff, couples DDPM (Ho et al., 2020b) and Discrete Masked Diffusion (Shi et al., 2024a; Sahoo et al., 2024) to synthesize tabular data.

## 3. Tabular Diffusion Models

Unlike images that contain only a single data type, the heterogeneous nature of tabular data requires an intricate mechanism to model continuous and discrete features. We begin with the following preambles where we denote a tabular dataset as  $\mathcal{D} = \{\mathbf{z}^{(i)}\}_{i=1}^N = \{[\mathbf{x}^{(i)}, \mathbf{c}^{(i)}]\}_{i=1}^N$ , where  $N$  represents the number of samples (rows). Each sample consists of continuous (numerical) features  $\mathbf{x}^{(i)} \in \mathbb{R}^{D_{\text{cont}}}$  and discrete (categorical) features  $\mathbf{c}^{(i)} \in \prod_{j \in \{1, \dots, D_{\text{cat}}\}} \{1, \dots, K_j\}$ , where  $D_{\text{cont}}$  is the number of continuous features,  $D_{\text{cat}}$  is the number of categorical features, and  $K_j$  is the number of unique categories for the  $j$ -th categorical feature.

### 3.1. Tabular Diffusion Framework

Traditional Diffusion Models (Ho et al., 2020b; Song & Ermon, 2020; Sohl-Dickstein et al., 2015) operate under a Markovian noising and denoising process where the forward noising iteratively corrupts the data distribution into a Gaussian distribution via Gaussian convolution, and the reverse denoising learns to transform Gaussian noise back into meaningful data samples. Due to the multi-modality of tabular data, tabular diffusion models have been designed to handle each tabular column as its innate continuous and discrete data type. The joint continuous-discrete forward diffusion process is defined as:

$$q(\mathbf{z}_t | \mathbf{z}_0) = q(\mathbf{x}_t | \mathbf{x}_0) \cdot q(\mathbf{c}_t | \mathbf{c}_0). \quad (1)$$

The reverse joint diffusion process parameterized by  $\theta$  is learned and given by:

$$p_\theta(\mathbf{z}_{t-1} | \mathbf{z}_t) = p_\theta(\mathbf{x}_{t-1} | \mathbf{x}_t, \mathbf{c}_t) \cdot p_\theta(\mathbf{c}_{t-1} | \mathbf{x}_t, \mathbf{c}_t). \quad (2)$$

At a high level, the latest state-of-the-art tabular diffusion models (Shi et al., 2024b; Kotelnikov et al., 2023; Lee et al., 2023) are trained on a separate continuous-discrete data representation via continuous Gaussian diffusion (Ho et al., 2020b; Karras et al., 2022) and discrete diffusion (Shi et al., 2024a; Hoogeboom et al., 2021).

**Gaussian Diffusion of Continuous Features.** The diffusion process of continuous features can be formulated using discrete-time Gaussian diffusion (Ho et al., 2020b; Nichol & Dhariwal, 2021); the noising process is described as:

$$q(\mathbf{x}_t | \mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t | \sqrt{\alpha_t^x} \mathbf{x}_{t-1}, (1 - \alpha_t^x) \mathbf{I}), \quad (3)$$

where  $\alpha_t^x$  is a hyperparameter to control the magnitude of the noising process. Per Tweedie’s Lemma (Efron, 2011; Robbins et al., 1992) indicates that the mean of denoising distribution can be obtained by the score of the distribution:

$$\mathbb{E}_q[\mathbf{x}_{t-1} | \mathbf{x}_t] = \frac{1}{\sqrt{\alpha_t^x}} \left( \mathbf{x}_t - \frac{1 - \alpha_t^x}{\sqrt{1 - \alpha_t^x}} \nabla_{\mathbf{x}_t} \log q(\mathbf{x}_t) \right), \quad (4)$$

in which we can use the approximated score function  $\nabla_{\mathbf{x}} \log p_\theta(\mathbf{x}) \approx \nabla_{\mathbf{x}} \log q(\mathbf{x})$  learned using denoising score matching (Song & Ermon, 2020; Vincent, 2011). This yields the denoising loss function for continuous features:

$$\mathcal{L}_{\mathbf{x}} = \mathbb{E}_{t, \mathbf{x}_0, \mathbf{x}_t, \mathbf{c}_t} \left[ \|\nabla_{\mathbf{x}_t} \log q(\mathbf{x}_t | \mathbf{x}_0) - \nabla_{\mathbf{x}_t} \log p_\theta(\mathbf{x}_t, \mathbf{c}_t)\|^2 \right]. \quad (5)$$

**Discrete Diffusion of Categorical Features.** Generating categorical features can be achieved using discrete-time diffusion processes tailored to handle discrete data (Hoogeboom et al., 2021; Campbell et al., 2022). The noising process for categorical variables is defined by transitioning between categorical states over discrete timesteps. The reverse process aims to learn the transition probabilities.

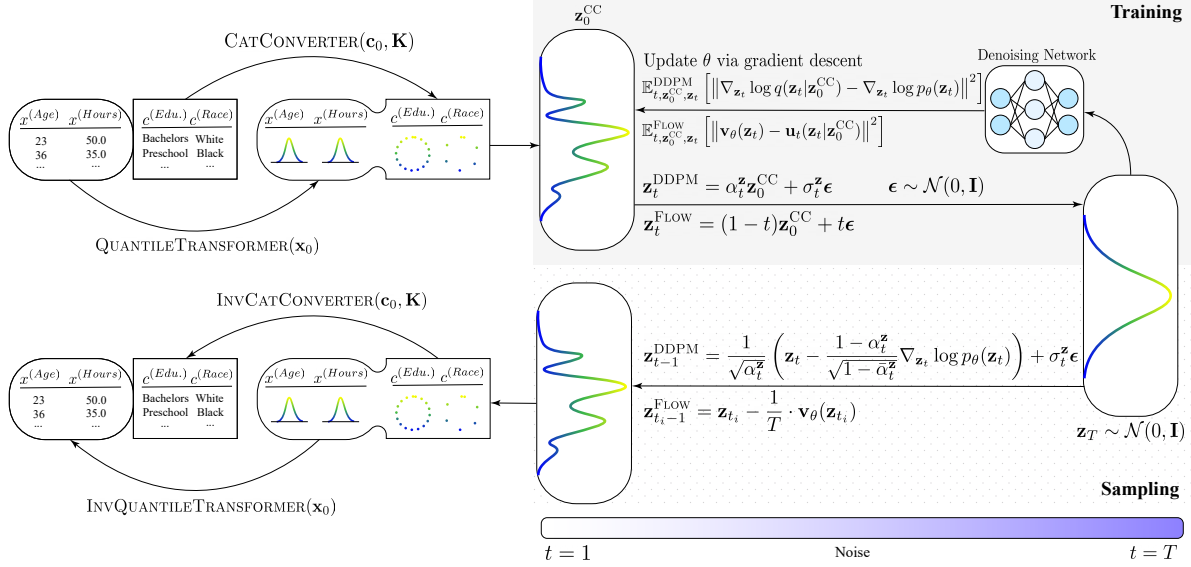


Figure 1. The TABGEN Architecture. TABGEN transforms and unifies the data space under a continuous regime via the CATCONVERTER. A diffusion or flow matching process is trained to optimize the denoising network. Once training is completed, samples can be generated through a reverse denoising process before inverse transforming back into their original data representation.

**Unified Continuous Diffusion.** Training diffusion models on a separate continuous-discrete data representation comes with the challenge of joint optimization. An alternative is to model a unified continuous data representation that circumvents these challenges. Kim et al. (2022) introduces STASY which utilizes a naive one-hot encoding representation to unify the data space but necessitates additional computationally heavy tricks to improve its performance. STASY trains a diffusion model on a subset of data before gradually extending to the whole training set. Thus, in addition to the diffusion model parameter,  $\theta$ , the model is required to learn a selection importance vector,  $\mathbf{v} \in [0, 1]$  through an alternative convex search (Bazarra et al., 1993). The search solves the loss function by iteratively optimizing between  $\theta$  and  $\mathbf{v}$ . The training objective  $\mathcal{L}_{\mathbf{x}, \mathbf{v}}^{\text{STASY}}$  is as follows:

$$\mathbf{v} \cdot \mathbb{E}_{t, \mathbf{x}_0, \mathbf{x}_t, \mathbf{c}_t} [\|\nabla_{\mathbf{x}_t} \log q(\mathbf{x}_t | \mathbf{x}_0) - \log p_\theta(\mathbf{x}_t, \mathbf{c}_t)\|^2] + r(\mathbf{v}; \alpha, \beta), \quad (6)$$

where  $r(\cdot)$  is a self-paced regularizer that requires fine-tuning. Another method by Zhang et al. (2023) introduces TABSYN that unifies the data space by transforming discrete features using a learned  $\beta$ -VAE latent embedding. The architecture of the VAE consists of a tokenizer that encodes discrete features into a one-hot vector, then parameterizes each category as a learnable vector learned by a Transformer (Vaswani, 2017) encoder. Likewise to STASY, TABSYN is also computationally expensive since it relies on training a separate VAE before the diffusion process to obtain latent codes. Additionally, the quality of synthetic data is highly dependent on the quality of the latent space. In the following section, we show that leveraging our simple and effective continuous data representation to train tabular

diffusion models is essential in synthesizing high-quality tabular data.

## 4. Method

Tabular data must undergo preprocessing preceding the diffusion generative process, and we find that transforming them into intelligible representations streamlines and improves the training of diffusion models.

### 4.1. Geometric Implications on the Data Representation

**Dense and Compact Representation.** A purely sparse representation, such as one-hot encoding, suffers from the *curse of dimensionality* (Bellman, 1957; Goodfellow et al., 2016), where the feature space grows exponentially with the number of categories, reducing model generalization (Krishnan et al., 2017; Poslavskaya & Korolev, 2023). Diffusion models, which learn to generate data through iterative perturbations and reconstructions of noise, encounter geometric challenges when applied to high-dimensional representations of categorical features. In this discussion, we provide a geometric analysis of these challenges using one-hot.

Let  $\{e_1, e_2, \dots, e_K\} \subset \mathbb{R}^K$  denote the set of standard basis vectors, where

$$e_k = (0, \dots, 0, \underbrace{1}_{k\text{-th entry}}, 0, \dots, 0), \quad (7)$$

represents the  $k$ -th category in a one-hot setting. For any point  $x \in \mathbb{R}^K$ , let

$$D_k(x) = \|x - e_k\|_2 \quad (8)$$

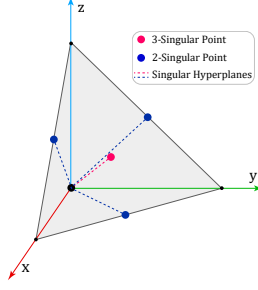


Figure 2. Singular Regions in a 3D One-Hot Setting.

denote the Euclidean distance between  $x$  and  $e_k$ . On the data manifold, there exist “singular” points where the vector field is hard to learn for diffusion models with Gaussian transitions. We demonstrate this with a uniform  $K$ -categorical distribution. Specifically, we define:

**Definition 4.1** (*n-singular point*). A point  $x \in \mathbb{R}^K$  is an  $n$ -singular point if there exists a subset  $\mathcal{S} \subseteq \{1, \dots, K\}$  with  $|\mathcal{S}| = n$  such that:

1.  $D_k(x) = D_{k'}(x), \quad \forall k, k' \in \mathcal{S}.$
2.  $D_k(x) \neq D_m(x), \quad \forall k \in \mathcal{S}, \forall m \notin \mathcal{S}.$

An  $n$ -singular point can be extended as a *minimal*  $n$ -singular point if it satisfies Definition 4.1, and minimizes the Euclidean distance to all basis vectors in  $\mathcal{S}$ :  $\min_x \|x - e_k\|_2 \quad \forall k \in \mathcal{S}$ . Hence, the minimal  $n$ -singular point is given by:

$$x_{\mathcal{S}}^{(n)} = \frac{1}{n} \sum_{k \in \mathcal{S}} e_k, \quad (9)$$

that corresponds to the centroid of the  $n$  basis vectors.

**Definition 4.2** (*n-singular hyperplane*). For each minimal  $n$ -singular point, there exists an *n-singular hyperplane* that is comprised of the set of all  $n$ -singular points associated with its respective  $n$  basis vectors in  $\mathcal{S}$ . Formally, it is defined as:

$$H_{\mathcal{S}} = \{x \in \mathbb{R}^K \mid D_k(x) = D_{k'}(x), \quad \forall k, k' \in \mathcal{S}\}, \quad (10)$$

where  $H_{\mathcal{S}}$  is an affine subspace in  $\mathbb{R}^K$  of dimension  $\dim(H_{\mathcal{S}}) = K - |\mathcal{S}| + 1$ . The hyperplane spans the corresponding minimal  $n$ -singular point  $x_{\mathcal{S}}^{(n)}$  and non-minimal  $n$ -singular points. For each  $n < K$ , there are  $\binom{K}{n}$  distinct  $n$ -singular hyperplanes, one for each subset  $\mathcal{S}$  of size  $n$ . Across all  $2 \leq n \leq K$ , the number of minimal  $n$ -singular points on the probability simplex scales combinatorially:  $\sum_{n=2}^K \binom{K}{n} = 2^K - (K + 1)$ . Thus, each minimal singular point carries the additional complexity of a continuous singular hyperplane.

Diffusion models rely on gradients derived from a learned vector field to denoise data iteratively. For regions within

proximity of the singular hyperplanes, gradients suffer from high variance due to conflicting directions arising from equidistant basis points. In non-uniform cases like tabular datasets, the distance between singular points and one-hot vectors should be weighted by category probabilities and the likelihood of Gaussian perturbations. Singular points may persist with significant probability, creating “voids” in the embedding space where diffusion models face high variance in learning signals. Additionally, sparsity in tabular datasets contributes further to high variance, making it challenging to learn tabular diffusion models with one-hot encoding.

In Figure 2, we depict a three-dimensional setting of the one-hot representation. As illustrated, a minimal 3-singular point occurs at the (red) centroid,  $(\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$ . Along with the non-minimal 3-singular points, these points form a (red-dashed) 1-dimensional singular hyperplane  $H_{\{1,2,3\}}$  perpendicular to the centroid of the probability simplex formed by  $e_1, e_2, e_3$ . Similarly, there are  $\binom{3}{2}$  2-singular (blue) points accompanied by their respective (blue-dashed) hyperplanes  $H_{\{1,2\}}, H_{\{1,3\}}$ , and  $H_{\{2,3\}}$ , each of which is a 2-dimensional affine subspace (plane). These singular hyperplanes pose more difficulty for diffusion models to learn effectively, especially when  $K$  is large.

**Separability.** Categories are mapped to distinct and separable representations, enabling the neural network to learn decision boundaries in the continuous embedding space if necessary (Bengio et al., 2013). This property is essential for the generative process to distinguish between different categories effectively for nominal categorical features. While higher-dimensional spaces admit better separability, they naturally introduce higher-order singular hyperplanes that complicate training. On the other hand, if the representations are packed too closely, small noise perturbations could make them collapse into a single category. This presents a density-separability trade-off. Heuristically, sparsity has a significant impact on harming generative performance. Thus, we aim to reduce dimensionality while maintaining sufficient separation to preserve categorical distinctions.

## 4.2. CATCONVERTER

Inspired by the discrete Fourier transform (DFT) (Oppenheim, 1999; Bracewell & Kahn, 1966), we draw on the concept of roots of unity to design a continuous representation for categorical variables, the CATCONVERTER. Notably, our method is plug-and-play, which does not require additional computation overhead. In harmonic analysis, the DFT map signals into the frequency domain using complex exponentials, where the  $K$ -th roots of unity represent equally spaced points on the unit circle, given by phases:

$$\theta_k = \frac{2\pi k}{K_j^{(i)}} \quad \text{for } k = 0, 1, \dots, K_j^{(i)} - 1. \quad (11)$$



Table 1. Dimensionality of categorical representations.

ENCODING SCHEMES	DIMENSIONS
ONE-HOT	$\sum_{j=1}^{D_{\text{CAT}}} K_j$
LEARNED EMBEDDING	$d_{\text{EMB}} \cdot D_{\text{CAT}}$
ANALOG BITS	$\sum_{j=1}^{D_{\text{CAT}}} \lceil \log_2(K_j) \rceil$
DICTIONARY	$D_{\text{CAT}}$
CATCONVERTER	$2 \cdot D_{\text{CAT}}$

Table 2. Average AUC/RMSE gap of unified (TABGEN-DDPM) vs. separated representation (TABDDPM) to the real data.

METHODS	AVERAGE GAP
TABDDPM	10.34%
TABGEN-DDPM	<b>3.08%</b>

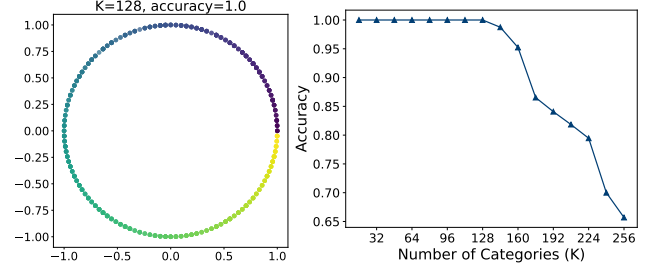
Analogously, we treat a categorical feature  $c_j^{(i)}$  with  $K_j^{(i)}$  distinct values as selecting one of these  $K_j^{(i)}$  points on the unit circle. Each category is thus mapped to a unique phase, and we represent it using the real and imaginary components of the corresponding complex exponential:

$$\text{CATCONVERTER}(c_j^{(i)}, K_j^{(i)}) = \left[ \cos\left(\frac{2\pi c_j^{(i)}}{K_j^{(i)}}\right), \sin\left(\frac{2\pi c_j^{(i)}}{K_j^{(i)}}\right) \right], \quad (12)$$

where  $\text{CATCONVERTER}() \in \mathbb{R}^{2 \cdot D_{\text{CAT}}}$ . Viewing a categorical entry as a discrete harmonic index enables us to embed the category in a two-dimensional phase space. This viewpoint retains the fundamental DFT insight—namely, that evenly spaced phase points around the unit circle furnish a natural orthonormal coordinate system—while enabling a smooth transition from discrete categories to a dense and continuous geometry-aware representation.

For CATCONVERTER, there exists a minimal  $K$ -singular point. Despite that, and some 2-singular point sets, there are no  $n$  singular points or hyperplanes for  $2 < n < K$ . Therefore, any  $n$ -singular point for  $n > 2$  coincides with the minimal  $K$ -singular point in this 2D representation.

Methods like Analog Bits (Chen et al., 2022) admit a similar idea of shrinking data dimension at log scale while CATCONVERTER offers an even denser 2-dimensional representation, as shown in Table 1. However, a potential concern is the ability to represent nominal features where representations like one-hot intuitively assign each category to each dimension, guaranteeing separability. We show that CATCONVERTER’s representation has ample separability for handling high-cardinality nominal categorical features commonly found in conventional tabular datasets. As illustrated in Figure 3, our representation could be easily distinguished by a small MLP among 128 categories mapped onto a continuous phase space. In addition to nominal features, the separability of CATCONVERTER coupled with its circular geometry naturally accommodates both periodic and ordinal


 Figure 3. Separability of CATCONVERTER. CATCONVERTER preserves *nominal* representations for up to 128 categories.

features. This facilitates an enhanced preservation of the feature’s intrinsic nature and characteristics. Note that prior to CATCONVERTER, we explore and introduce an intuitive static one-dimensional embedding, Dictionary (DIC) but we find that it underperforms.

### 4.3. TABGEN

In our paper, we model a unified continuous diffusion process with DDPM (Ho et al., 2020b) or Flow Matching (FM) (Lipman et al., 2022). To unify the dataspace, we represent discrete variables via CATCONVERTER( $c_0, K$ ) and concatenate with continuous features forming our dataset. Our dataset can be denoted as  $\{z_0^{\text{CC}}\} = \{[x_0, c_0^{\text{CC}}]\}$ .

For DDPM (Ho et al., 2020b), we define the forward process by progressively perturbing the data distribution using a Gaussian noise model, where the latent state at time  $t$ ,  $z_t$ , is computed as  $z_t = \alpha_t^z z_0^{\text{CC}} + \sigma_t^z \epsilon$  with  $\epsilon \sim \mathcal{N}(0, \mathbf{I})$ . The denoising model  $p_\theta(z_t)$  is trained to predict the posterior gradients  $\nabla_{z_t} \log q(z_t | z_0^{\text{CC}})$ , minimizing the weighted variance loss  $\mathcal{L}_{\text{TABGEN-DDPM}}$ :

$$\mathbb{E}_{t, z_0^{\text{CC}}, z_t} \left[ \left\| \nabla_{z_t} \log q(z_t | z_0^{\text{CC}}) - \nabla_{z_t} \log p_\theta(z_t) \right\|^2 \right]. \quad (13)$$

For FM (Lipman et al., 2022), we instead define the dynamics in terms of a conditional vector field  $u_t(z_t | z_0^{\text{CC}}) = \epsilon - z_0^{\text{CC}}$  with  $z_t := (1 - t)z_0^{\text{CC}} + t\epsilon$ . The model learns the target field by minimizing the discrepancy between the predicted vector  $v_\theta(z_t)$  and the ground truth  $u_t(z_t | z_0^{\text{CC}})$  through the flow matching loss  $\mathcal{L}_{\text{TABGEN-FLOW}}$ :

$$\mathbb{E}_{t, z_0^{\text{CC}}, z_t} \left[ \left\| v_\theta(z_t) - u_t(z_t | z_0^{\text{CC}}) \right\|^2 \right]. \quad (14)$$

At sampling time, TABGEN-DDPM performs reverse diffusion by iteratively denoising  $z_t$  back to  $z_0^{\text{CC}}$ , while TABGEN-FLOW solves a deterministic ordinary differential equation (ODE) of the vector field. Our complete training and sampling algorithms can be found in Figure 4.

To conclude, we examine the average gap across all datasets in downstream tabular generation quality of a unified CATCONVERTER representation, TABGEN-DDPM, and a separated representation, TABDDPM, to the real data. In Table

**Algorithm 1** Training TABGEN-DDPM/FLOW

<b>DDPM:</b> 2: <b>while</b> not converged <b>do</b> 3:   Sample $\mathbf{z}_0 = [\mathbf{x}_0, \mathbf{c}_0] \sim p(\mathbf{z})$ 4:   Encode $\mathbf{c}_0^{\text{CC}} \leftarrow \text{CATCONVERTER}(\mathbf{c}_0, \mathbf{K})$ 5:   Encode $\mathbf{x}_0 \leftarrow \text{QUANTILETRANSFORMER}(\mathbf{x}_0)$ 6: $\mathbf{z}_0^{\text{CC}} \leftarrow \text{CONCAT}(\mathbf{x}_0, \mathbf{c}_0^{\text{CC}})$ 7:   Sample $t \sim \text{Uniform}(\{1, \dots, T\})$ 8:   Sample noise $\epsilon \sim \mathcal{N}(0, \mathbf{I})$ 9:   Compute $\mathbf{z}_t = \alpha_t^z \mathbf{z}_0^{\text{CC}} + \sigma_t^z \epsilon$  Compute $\mathcal{L}_{\text{TABGEN-DDPM}} =$ $\mathbb{E}_{t, \mathbf{z}_0^{\text{CC}}, \mathbf{z}_t} [\ \nabla_{\mathbf{z}_t} \log q(\mathbf{z}_t   \mathbf{z}_0^{\text{CC}}) - \nabla_{\mathbf{z}_t} \log p_\theta(\mathbf{z}_t)\ ^2]$ Update $\theta$ via gradient descent $\nabla_\theta \mathcal{L}_{\text{TABGEN-DDPM}}$	<b>FLOW MATCHING:</b>  Compute $\mathbf{z}_t = (1 - t)\mathbf{z}_0^{\text{CC}} + t\epsilon$ Define $\mathbf{u}_t(\mathbf{z}_t   \mathbf{z}_0^{\text{CC}}) = \epsilon - \mathbf{z}_0^{\text{CC}}$ Compute $\mathcal{L}_{\text{TABGEN-FLOW}} =$ $\mathbb{E}_{t, \mathbf{z}_0^{\text{CC}}, \mathbf{z}_t} [\ \mathbf{v}_\theta(\mathbf{z}_t) - \mathbf{u}_t(\mathbf{z}_t   \mathbf{z}_0^{\text{CC}})\ ^2]$ Update $\theta$ via gradient descent $\nabla_\theta \mathcal{L}_{\text{TABGEN-FLOW}}$
--	--

 10: **end while**
**Algorithm 2** Sampling TABGEN-DDPM/FLOW

<b>DDPM:</b> 2: Sample $\mathbf{z}_T^{\text{CC}} \sim \mathcal{N}(0, \mathbf{I})$ 3: <b>for</b> $t = T, \dots, 1$ <b>do</b> 4:   Sample $\epsilon \sim \mathcal{N}(0, \mathbf{I})$ if $t > 1$ , else $\epsilon = 0$  $\mathbf{z}_{t-1} = \frac{1}{\sqrt{\alpha_t^z}} \left( \mathbf{z}_t - \frac{1 - \alpha_t^z}{\sqrt{1 - \alpha_t^z}} \nabla_{\mathbf{z}_t} \log p_\theta(\mathbf{z}_t) \right) + \sigma_t^z \epsilon$  5: <b>end for</b> 6: Output $\mathbf{z}_0^{\text{CC}} = [\mathbf{x}_0, \mathbf{c}_0^{\text{CC}}]$ 7: Decode $\mathbf{c}_0 \leftarrow \text{INVCATCONVERTER}(\mathbf{c}_0^{\text{CC}}, \mathbf{K})$ 8: Decode $\mathbf{x}_0 \leftarrow \text{INVQUANTILETRANSFORMER}(\mathbf{x}_0)$ 9: $\mathbf{z}_0 \leftarrow \text{CONCAT}(\mathbf{x}_0, \mathbf{c}_0)$ 10: <b>return</b> $\mathbf{z}_0^{\text{CC}}$	<b>FLOW MATCHING:</b>  Discretize time $t_i = i/T$ ,      for $i = T, T - 1, \dots, 1$  $\mathbf{z}_{t_i-1} = \mathbf{z}_{t_i} - \frac{1}{T} \cdot \mathbf{v}_\theta(\mathbf{z}_{t_i})$ ,      via Euler ODE Solver
--	---

Figure 4. Training and sampling algorithms of TABGEN-DDPM/FLOW.

2, the results demonstrate that the generation quality of DDPMs improves significantly using our CATCONVERTER unified data representation, emphasizing our effectiveness.

## 5. Experiments

We evaluate the performance of TABGEN-DDPM and TABGEN-FLOW. Our results are compared against various datasets, baselines, and benchmarks.

### 5.1. Setup

**Implementation Information.** Experimental results are obtained over an average of 20 sampling seeds using the best-validated model. Continuous features are encoded using a QuantileTransformer (Pedregosa et al., 2011). The order in which the categories of a feature are assigned in our encoding schemes is based on lexicographic ordering for simplicity. Further details are in Appendix A.

**Datasets.** We select seven datasets from the UCI Machine Learning Repository to conduct our experiments. This in-

cludes Adult, Default, Shoppers, Stroke, Diabetes, Beijing, and News which contain a mix of continuous and discrete features. Further details are in Appendix B.1.

**Baselines.** We compare our model against existing diffusion baselines for tabular generation since they are the best-performing. This includes STASY (Kim et al., 2022), CoDi (Lee et al., 2023), TABDDPM (Kotelnikov et al., 2023), TABSYN (Zhang et al., 2023), and TABDIFF<sup>1</sup> (Shi et al., 2024b). Further details are in Appendix B.2.

**Benchmarks.** We observe that downstream task performance measured by machine learning efficiency (MLE) typically translates to most other benchmarks. Thus, the primary quality benchmark in our main paper will be MLE for brevity, and a privacy benchmark, membership inference attacks (MIA). The remaining fidelity benchmarks include: column-wise density (CWD), pairwise-column correlation (PCC),  $\alpha$ -precision,  $\beta$ -recall, and classifier-two-sample test

<sup>1</sup>No publicly available code for TABDIFF; results taken directly from TABDIFF.

Table 3. Ablation Study on TABGEN’s unified representation versus a separate representation.

METHODS	AUC $\uparrow$				RMSE $\downarrow$		
	ADULT	DEFAULT	SHOPPERS	STROKE	DIABETES	BEIJING	NEWS
TABDDPM	0.910 $\pm$ .001	0.761 $\pm$ .004	0.915 $\pm$ .004	0.808 $\pm$ .033	0.634 $\pm$ .007	0.592 $\pm$ .012	3.46 $\pm$ 1.25
TABGEN-DDPM	<b>0.913<math>\pm</math>.002</b>	<b>0.764<math>\pm</math>.005</b>	<b>0.926<math>\pm</math>.005</b>	<b>0.869<math>\pm</math>.027</b>	<b>0.636<math>\pm</math>.003</b>	<b>0.508<math>\pm</math>.006</b>	<b>0.836<math>\pm</math>.001</b>
TABFLOW	0.908 $\pm$ .002	0.742 $\pm$ .008	0.914 $\pm$ .005	0.821 $\pm$ .082	0.630 $\pm$ .005	0.574 $\pm$ .010	0.850 $\pm$ .017
TABGEN-FLOW	<b>0.912<math>\pm</math>.002</b>	<b>0.782<math>\pm</math>.005</b>	<b>0.919<math>\pm</math>.005</b>	<b>0.830<math>\pm</math>.028</b>	<b>0.636<math>\pm</math>.002</b>	<b>0.536<math>\pm</math>.006</b>	<b>0.814<math>\pm</math>.002</b>

Table 4. Ablation Study on categorical representations under a unified continuous data space.

METHODS	AUC $\uparrow$				RMSE $\downarrow$		
	ADULT	DEFAULT	SHOPPERS	STROKE	DIABETES	BEIJING	NEWS
ONEHOT-DDPM	0.476 $\pm$ .057	0.557 $\pm$ .052	0.799 $\pm$ .126	0.797 $\pm$ .133	0.484 $\pm$ .032	2.143 $\pm$ .339	0.840 $\pm$ .020
LEARNED1D-DDPM	0.611 $\pm$ .008	0.575 $\pm$ .012	0.876 $\pm$ .028	0.743 $\pm$ .032	0.460 $\pm$ .010	0.921 $\pm$ .006	0.858 $\pm$ .010
LEARNED2D-DDPM	0.793 $\pm$ .003	0.290 $\pm$ .009	0.103 $\pm$ .011	0.850 $\pm$ .035	0.516 $\pm$ .007	0.969 $\pm$ .005	0.857 $\pm$ .023
I2B-DDPM	0.911 $\pm$ .001	0.762 $\pm$ .003	0.919 $\pm$ .004	0.852 $\pm$ .029	0.632 $\pm$ .005	0.542 $\pm$ .008	0.844 $\pm$ .013
DIC-DDPM	0.912 $\pm$ .002	0.763 $\pm$ .003	0.910 $\pm$ .004	0.824 $\pm$ .027	<b>0.636<math>\pm</math>.006</b>	0.547 $\pm$ .008	0.851 $\pm$ .013
TABGEN-DDPM	<b>0.913<math>\pm</math>.002</b>	<b>0.764<math>\pm</math>.005</b>	<b>0.926<math>\pm</math>.005</b>	<b>0.869<math>\pm</math>.027</b>	0.634 $\pm$ .003	<b>0.508<math>\pm</math>.006</b>	<b>0.836<math>\pm</math>.001</b>
ONEHOT-FLOW	0.895 $\pm$ .003	0.759 $\pm$ .005	0.910 $\pm$ .006	0.812 $\pm$ .129	0.604 $\pm$ .035	0.765 $\pm$ .016	0.850 $\pm$ .017
LEARNED1D-FLOW	0.260 $\pm$ .014	0.438 $\pm$ .009	0.134 $\pm$ .015	0.142 $\pm$ .033	0.506 $\pm$ .007	0.806 $\pm$ .009	0.873 $\pm$ .007
LEARNED2D-FLOW	0.126 $\pm$ .012	0.709 $\pm$ .008	0.868 $\pm$ .007	0.180 $\pm$ .030	0.466 $\pm$ .008	0.787 $\pm$ .007	0.866 $\pm$ .005
I2B-FLOW	0.911 $\pm$ .001	0.763 $\pm$ .004	0.910 $\pm$ .005	0.797 $\pm$ .027	0.636 $\pm$ .003	0.543 $\pm$ .007	0.847 $\pm$ .014
DIC-FLOW	0.912 $\pm$ .002	0.763 $\pm$ .004	0.903 $\pm$ .005	0.807 $\pm$ .028	0.637 $\pm$ .007	0.561 $\pm$ .007	0.853 $\pm$ .014
TABGEN-FLOW	<b>0.912<math>\pm</math>.002</b>	<b>0.782<math>\pm</math>.005</b>	<b>0.919<math>\pm</math>.005</b>	<b>0.830<math>\pm</math>.028</b>	<b>0.637<math>\pm</math>.002</b>	<b>0.536<math>\pm</math>.006</b>	<b>0.814<math>\pm</math>.002</b>

(C2ST). Further details regarding benchmark information and additional results are in Appendix B.3 and C.

## 5.2. Ablation Studies

**Unified and Separate Data Representations.** We analyze the impact of training tabular diffusion models between our unified data representation and a separate data representation. In Table 3, we compare TABGEN-DDPM and TABGEN-FLOW to TABDDPM (Kotelnikov et al., 2023) and TABFLOW. Note that we introduce TABFLOW, by modeling continuous features using Flow Matching (Lipman et al., 2022; Liu et al., 2022) and categorical features using Discrete Flow Matching (Campbell et al., 2024) under the same separate continuous-discrete data representation as TABDDPM. Results illustrate that data synthesized by diffusion-based models using TABGEN’s unified representation consistently outperforms a separate representation when training diffusion and flow matching models.

**Categorical Representations.** We conduct ablation studies with respect to categorical data representations used in existing diffusion baselines to assess the performance of diffusion models under a unified data representation. This includes one-hot encoding used in Kim et al. (2022); Lee et al. (2023); Shi et al. (2024b), learned one-dimensional

and two-dimensional embeddings (Mikolov et al., 2013), and Analog Bits (I2B) (Chen et al., 2022) from the discrete image diffusion domain. Additionally, we introduce an intuitive static one-dimensional embedding, Dictionary (DIC). Results in Table 4 showcase that TABGEN outperforms the ablated representations in generating data under a unified data representation via diffusion models. Further details regarding the various representations are in Appendix A.1.

## 5.3. Baseline Performance

**Generation Quality.** We benchmark TABGEN against baselines on a downstream MLE task where we determine the AUC and RMSE of XGBoost (Chen & Guestrin, 2016) for classification and regression tasks on the generated synthetic datasets. The diffusion models are trained on the training set, and synthetic samples of equivalent size are generated, which are then evaluated using the testing set. As observed in Table 5, CATCONVERTER is effective on both DDPM and Flow Matching since TABGEN-DDPM and TABGEN-FLOW consistently attain the best MLE performance compared to existing baselines. Additionally, our method is the first to yield performance levels greater than the Real datasets including Default, Stroke, and News.

**Privacy Preservation.** One of the main applications of syn-

Table 5. AUC (classification) and RMSE (regression) scores of Machine Learning Efficiency. Higher scores indicate better performance.

METHODS	AUC $\uparrow$				RMSE $\downarrow$		
	ADULT	DEFAULT	SHOPPERS	STROKE	DIABETES	BEIJING	NEWS
REAL	0.927 $\pm$ .000	0.770 $\pm$ .005	0.926 $\pm$ .001	0.852 $\pm$ .002	0.642 $\pm$ .003	0.423 $\pm$ .003	0.842 $\pm$ .002
STASY	0.906 $\pm$ .001	0.752 $\pm$ .006	0.914 $\pm$ .005	0.833 $\pm$ .030	0.629 $\pm$ .003	0.656 $\pm$ .014	0.871 $\pm$ .002
CoDI	0.871 $\pm$ .006	0.525 $\pm$ .006	0.865 $\pm$ .006	0.798 $\pm$ .032	0.469 $\pm$ .009	0.818 $\pm$ .021	1.21 $\pm$ .005
TABDDPM	0.910 $\pm$ .001	0.761 $\pm$ .004	0.915 $\pm$ .004	0.808 $\pm$ .033	0.634 $\pm$ .007	0.592 $\pm$ .012	3.46 $\pm$ 1.25
TABSYN	0.906 $\pm$ .001	0.755 $\pm$ .004	0.918 $\pm$ .004	0.845 $\pm$ .035	0.634 $\pm$ .003	0.586 $\pm$ .013	0.862 $\pm$ .021
TABDIFF	0.912 $\pm$ .002	0.763 $\pm$ .005	0.921 $\pm$ .004	—	—	0.555 $\pm$ .013	0.866 $\pm$ .021
TABGEN-DDPM	<b>0.913<math>\pm</math>.002</b>	0.764 $\pm$ .005	<b>0.926<math>\pm</math>.005</b>	<b>0.869<math>\pm</math>.027</b>	0.636 $\pm$ .003	<b>0.508<math>\pm</math>.006</b>	0.836 $\pm$ .001
TABGEN-FLOW	0.912 $\pm$ .002	<b>0.782<math>\pm</math>.005</b>	0.919 $\pm$ .005	0.854 $\pm$ .028	<b>0.637<math>\pm</math>.002</b>	0.536 $\pm$ .006	<b>0.814<math>\pm</math>.002</b>

Table 6. Precision Scores of Membership Inference Attacks. A score closer to 50% is better for privacy-preservation.

METHODS	ADULT	DEFAULT	SHOPPERS	STROKE	DIABETES	BEIJING	NEWS
STASY	49.52 $\pm$ 0.50	51.84 $\pm$ 0.69	51.23 $\pm$ 2.38	54.22 $\pm$ 1.14	49.47 $\pm$ 0.22	48.98 $\pm$ 0.78	49.68 $\pm$ 0.70
CoDI	55.00 $\pm$ 7.26	46.66 $\pm$ 2.54	59.67 $\pm$ 11.74	54.54 $\pm$ 1.60	00.00 $\pm$ 0.00	39.19 $\pm$ 6.35	40.00 $\pm$ 24.49
TABDDPM	51.30 $\pm$ 0.07	51.34 $\pm$ 0.49	51.24 $\pm$ 1.48	53.45 $\pm$ 2.89	50.40 $\pm$ 0.65	50.43 $\pm$ 0.58	50.72 $\pm$ 0.97
TABSYN	50.91 $\pm$ 0.16	50.80 $\pm$ 0.56	51.00 $\pm$ 1.08	51.11 $\pm$ 1.54	50.10 $\pm$ 0.34	51.31 $\pm$ 0.42	49.86 $\pm$ 0.75
TABGEN-DDPM	<b>50.44<math>\pm</math>0.83</b>	50.07 $\pm$ 0.41	<b>49.86<math>\pm</math>0.98</b>	51.74 $\pm$ 1.85	<b>50.07<math>\pm</math>0.37</b>	51.02 $\pm$ 0.42	<b>49.87<math>\pm</math>0.99</b>
TABGEN-FLOW	50.65 $\pm$ 0.20	<b>50.04<math>\pm</math>0.95</b>	51.38 $\pm$ 1.66	<b>50.66<math>\pm</math>1.77</b>	50.24 $\pm$ 0.37	<b>50.35<math>\pm</math>0.60</b>	50.90 $\pm$ 0.93

Table 7. Training and Sampling Duration in Seconds.

METHODS	TRAINING	SAMPLING	TOTAL
STASY	6608.84	12.93	6621.77
CoDI	24039.96	9.41	24049.37
TABDDPM	3112.07	66.82	3178.89
TABSYN	2373.98 + 1084.82	10.54	3469.30
TABDIFF	5640	15.2	5655.2
TABGEN-DDPM	2070.59	59.00	2130.59
TABGEN-FLOW	2028.33	3.07	<b>2031.40</b>

thetic tabular data is privacy-preserving machine learning, where synthetic data is generated to protect sensitive information. We perform membership inference attacks (MIAs) to evaluate the vulnerability of the methods to privacy leakage (Shokri et al., 2017). A privacy-preserving model would yield precision scores of 50%, indicating that the attack is no better than random guessing. As demonstrated in Table 6, our method preserves privacy with MIAs scoring close to 50% for precision. Recall scores can be found in Appendix C.1 which delivers similar privacy preservation insights.

**Generation Efficacy.** Training and Sampling TABGEN is the most efficient among all baselines and does not necessitate additional computing power. We compare the generation efficacy using the Adult dataset with a balanced mix of continuous and discrete features. In Table 7, we observe that our method is the quickest to train and sample in duration. Furthermore, we also conducted experiments on the convergence speed for the training process. As illustrated in Figure 5a, our method converges to a high AUC earliest in

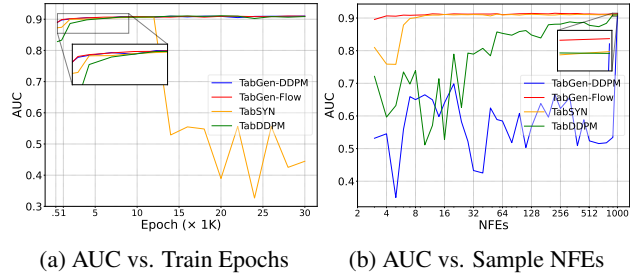


Figure 5. Training and Sampling Efficacy.

the training stages. Lastly, we assess the number of function evaluations (NFEs) the models take to sample. Figure 5b indicates that TABGEN-FLOW can attain its best performance as early as 8 NFEs. At 1000 NFEs, we observe that both TABGEN models are the best-performing.

## 6. Conclusion

In this work, we present TABGEN a tabular diffusion architecture trained with a simple yet effective unified continuous representation. Motivated by geometric implications on the data manifold, our representation is dense, separable, and captures meaningful information for diffusion models. We conducted extensive experiments on a myriad of datasets, baselines, and benchmarks to evaluate TABGEN. The results showcase TABGEN’s prowess in generating high-quality, privacy-preserving synthetic data while remaining computationally inexpensive.



## Impact Statement

This paper introduces TABGEN to generate high-quality synthetic tabular data while preserving privacy. This has the potential to benefit society by enabling data sharing in sensitive domains (e.g., healthcare, finance) where privacy constraints often restrict access to real-world data. Through its unified representation and diffusion-based methodology, TABGEN can help democratize data availability, support model development for underrepresented populations, and spur innovation in areas otherwise hindered by data scarcity. Nonetheless, there are risks. If the training data is biased, TABGEN may inadvertently reproduce or amplify existing bias in the synthetic samples. Additionally, while TABGEN strives to protect privacy, improper use or careless deployment might still enable adversaries to glean sensitive patterns. Finally, malicious actors could exploit synthetic data generation to produce fraudulent data or fabricate misleading results. These concerns underscore the importance of implementing robust ethical guidelines and validation procedures around synthetic data usage, ensuring that the benefits of TABGEN can be fully realized responsibly.

## References

- Alaa, A. M., van Breugel, B., Saveliev, E., and van der Schaar, M. How faithful is your synthetic data? sample-level metrics for evaluating and auditing generative models, 2022.
- Bazaraa, M. S., Sherali, H. D., and Shetty, C. M. *Nonlinear Programming: Theory and Algorithms*. John Wiley & Sons, New York, 1993.
- Bellman, R. *Dynamic Programming*. Princeton University Press, Princeton, NJ, 1957.
- Bengio, Y., Courville, A., and Vincent, P. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828, 2013. doi: 10.1109/TPAMI.2013.50.
- Bengio, Y., Courville, A., and Vincent, P. Representation learning: A review and new perspectives, 2014. URL <https://arxiv.org/abs/1206.5538>.
- Bracewell, R. and Kahn, P. B. The fourier transform and its applications. *American Journal of Physics*, 34(8):712–712, 1966.
- Campbell, A., Benton, J., Bortoli, V. D., Rainforth, T., Deligiannidis, G., and Doucet, A. A continuous time framework for discrete denoising models, 2022. URL <https://arxiv.org/abs/2205.14987>.
- Campbell, A., Yim, J., Barzilay, R., Rainforth, T., and Jaakkola, T. Generative flows on discrete state-spaces: Enabling multimodal flows with applications to protein co-design, 2024.
- Chen, T. and Guestrin, C. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD ’16. ACM, August 2016. doi: 10.1145/2939672.2939785. URL <http://dx.doi.org/10.1145/2939672.2939785>.
- Chen, T., Zhang, R., and Hinton, G. Analog bits: Generating discrete data using diffusion models with self-conditioning. *arXiv preprint arXiv:2208.04202*, 2022.
- Clore, J., Cios, K., DeShazo, J., and Strack, B. Diabetes 130-US hospitals for years 1999-2008. UCI Machine Learning Repository, 2014. DOI: <https://doi.org/10.24432/C5230J>.
- Datta, A. Us health insurance dataset, Feb 2020. URL <https://www.kaggle.com/datasets/teertha/ushealthinsurancedataset>.
- Dua, D. and Graff, C. Uci machine learning repository, 2017. URL <http://archive.ics.uci.edu/ml>.
- Efron, B. Tweedie’s formula and selection bias. *Journal of the American Statistical Association*, 106(496):1602–1614, 2011. ISSN 01621459. URL <http://www.jstor.org/stable/23239562>.
- Goodfellow, I., Bengio, Y., and Courville, A. *Deep Learning*. MIT Press, Cambridge, MA, 2016.
- Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., and Lerchner, A. beta-VAE: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations*, 2017. URL <https://openreview.net/forum?id=Sy2fzU9gl>.
- Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 6840–6851. Curran Associates, Inc., 2020a. URL [https://proceedings.neurips.cc/paper\\_files/paper/2020/file/4c5bcfec8584af0d967f1ab10179ca4b-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/2020/file/4c5bcfec8584af0d967f1ab10179ca4b-Paper.pdf).
- Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020b.
- Hoogetboom, E., Nielsen, D., Jaini, P., Forré, P., and Welling, M. Argmax flows and multinomial diffusion: Learning categorical distributions, 2021.

- Jolicoeur-Martineau, A., Fatras, K., and Kachman, T. Generating and imputing tabular data via diffusion and flow-based gradient-boosted trees, 2024.
- Karras, T., Aittala, M., Aila, T., and Laine, S. Elucidating the design space of diffusion-based generative models, 2022.
- Kim, J., Lee, C., and Park, N. Stasy: Score-based tabular data synthesis. *arXiv preprint arXiv:2210.04018*, 2022.
- Kingma, D. P. and Welling, M. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.
- Kotelnikov, A., Baranchuk, D., Rubachev, I., and Babenko, A. Tabddpm: Modelling tabular data with diffusion models. In *International Conference on Machine Learning*, pp. 17564–17579. PMLR, 2023.
- Krishnan, R. G., Liang, D., and Hoffman, M. On the challenges of learning with inference networks on sparse, high-dimensional data, 2017.
- Lautrup, A. D., Hyrup, T., Zimek, A., and Schneider-Kamp, P. Syntheval: a framework for detailed utility and privacy evaluation of tabular synthetic data. *Data Mining and Knowledge Discovery*, 39(1), December 2024. ISSN 1573-756X. doi: 10.1007/s10618-024-01081-4. URL <http://dx.doi.org/10.1007/s10618-024-01081-4>.
- Lee, C., Kim, J., and Park, N. Codi: Co-evolving contrastive diffusion models for mixed-type tabular synthesis. In *International Conference on Machine Learning*, pp. 18940–18956. PMLR, 2023.
- Lipman, Y., Chen, R. T., Ben-Hamu, H., Nickel, M., and Le, M. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.
- Liu, T., Qian, Z., Berrevoets, J., and van der Schaar, M. GOGGLE: Generative modelling for tabular data by learning relational structure. In *The Eleventh International Conference on Learning Representations*, 2023. URL <https://openreview.net/forum?id=fPVRcJqspu>.
- Liu, X., Gong, C., and Liu, Q. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022.
- Mikolov, T., Chen, K., Corrado, G., and Dean, J. Efficient estimation of word representations in vector space, 2013. URL <https://arxiv.org/abs/1301.3781>.
- Moro, S., Rita, P., and Cortez, P. Bank Marketing. UCI Machine Learning Repository, 2012. DOI: <https://doi.org/10.24432/C5K306>.
- Nichol, A. Q. and Dhariwal, P. Improved denoising diffusion probabilistic models. *arXiv preprint arXiv:2102.09672*, 2021.
- Onishi, S. and Meguro, S. Rethinking data augmentation for tabular data in deep learning. *arXiv preprint arXiv:2305.10308*, 2023.
- Oppenheim, A. V. *Discrete-time signal processing*. Pearson Education India, 1999.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
- Poslavska, E. and Korolev, A. Encoding categorical data: Is there yet anything ‘hotter’ than one-hot encoding?, 2023.
- Robbins, H. E., Kotz, S., and Johnson, N. L. *An Empirical Bayes Approach to Statistics*, pp. 388–394. Springer New York, New York, NY, 1992. ISBN 978-1-4612-0919-5. doi: 10.1007/978-1-4612-0919-5\_26. URL [https://doi.org/10.1007/978-1-4612-0919-5\\_26](https://doi.org/10.1007/978-1-4612-0919-5_26).
- Sahoo, S. S., Arriola, M., Schiff, Y., Gokaslan, A., Marroquin, E., Chiu, J. T., Rush, A., and Kuleshov, V. Simple and effective masked diffusion language models, 2024. URL <https://arxiv.org/abs/2406.07524>.
- Sauber-Cole, R. and Khoshgoftaar, T. M. The use of generative adversarial networks to alleviate class imbalance in tabular data: a survey. *Journal of Big Data*, 9(1):98, 2022.
- SDMetrics. Detection metrics (single table) - sdmetrics documentation, 2024. URL <https://docs.sdv.dev/sdmetrics/metrics/metrics-in-beta/detection-single-table>. Accessed: 2024-05-20.
- Shi, J., Han, K., Wang, Z., Doucet, A., and Titsias, M. K. Simplified and generalized masked diffusion for discrete data, 2024a. URL <https://arxiv.org/abs/2406.04329>.
- Shi, J., Xu, M., Hua, H., Zhang, H., Ermon, S., and Leskovec, J. Tabdiff: a multi-modal diffusion model for tabular data generation, 2024b. URL <https://arxiv.org/abs/2410.20626>.
- Shokri, R., Stronati, M., Song, C., and Shmatikov, V. Membership inference attacks against machine learning models, 2017. URL <https://arxiv.org/abs/1610.05820>.

- Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., and Ganguli, S. Deep unsupervised learning using nonequilibrium thermodynamics. In Bach, F. and Blei, D. (eds.), *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 2256–2265, Lille, France, 07–09 Jul 2015. PMLR. URL <https://proceedings.mlr.press/v37/sohl-dickstein15.html>.
- Song, Y. and Ermon, S. Generative modeling by estimating gradients of the data distribution, 2020. URL <https://arxiv.org/abs/1907.05600>.
- Tong, A., Malkin, N., Huguet, G., Zhang, Y., Rector-Brooks, J., Fatras, K., Wolf, G., and Bengio, Y. Improving and generalizing flow-based generative models with mini-batch optimal transport. In *ICML Workshop on New Frontiers in Learning, Control, and Dynamical Systems*, 2023.
- Vaswani, A. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.
- Vincent, P. A connection between score matching and denoising autoencoders. *Neural Computation*, 23(7):1661–1674, 2011. doi: 10.1162/NECO.a.00142.
- Xu, L., Skoularidou, M., Cuesta-Infante, A., and Veeramachaneni, K. Modeling tabular data using conditional gan, 2019.
- Xu, R., Baracaldo, N., and Joshi, J. Privacy-preserving machine learning: Methods, challenges and directions. *arXiv preprint arXiv:2108.04417*, 2021.
- Zhang, H., Zhang, J., Srinivasan, B., Shen, Z., Qin, X., Faloutsos, C., Rangwala, H., and Karypis, G. Mixed-type tabular data synthesis with score-based diffusion in latent space. *arXiv preprint arXiv:2310.09656*, 2023.

# Appendix

## Contents

<b>A Implementation</b>	<b>12</b>
A.1 Categorical Representations	12
A.2 Architecture	13
A.3 Hyperparameters	14
<b>B Experiments</b>	<b>14</b>
B.1 Datasets	14
B.2 Baselines	15
B.3 Benchmarks	16
<b>C Further Experimental Results</b>	<b>17</b>
C.1 Additional Baseline Results	18
C.2 Additional Ablation Results	19

## A. Implementation

The following delineates the foundation of our experiments:

- Codebase: Python & PyTorch
- CPU: AMD EPYC-Rome 7002
- GPU: NVIDIA A100 80GB PCIe

### A.1. Categorical Representations

**One-Hot Encoding.** The one-hot encoding (ONEHOT) representation of  $\mathbf{c}^{(i)}$  is constructed by concatenating the one-hot encoded vectors of each individual feature  $c_j^{(i)}$ . Specifically, the one-hot encoding for  $c_j^{(i)}$  is a vector  $\mathbf{e}(c_j^{(i)}) \in \{0, 1\}^{K_j}$ , where the  $k$ -th entry is defined as:

$$\mathbf{e}(c_j^{(i)})_k = \begin{cases} 1 & \text{if } k = c_j^{(i)}, \\ 0 & \text{otherwise,} \end{cases} \quad (15)$$

for  $k \in \{1, 2, \dots, K_j\}$ . The one-hot encoded representation of  $\mathbf{c}^{(i)}$  is then:

$$\text{ONEHOT}(\mathbf{c}^{(i)}) = [\mathbf{e}(c_1^{(i)}), \mathbf{e}(c_2^{(i)}), \dots, \mathbf{e}(c_{D_{\text{cat}}}^{(i)})]. \quad (16)$$

The resulting vector has a total length of  $\sum_{j=1}^{D_{\text{cat}}} K_j$ , which corresponds to the sum of the unique categories across all categorical features. A softmax or a logarithm is then applied to the one-hot representation to yield a continuous probability distribution.

**Learned Embeddings.** Learned Embeddings (LEARNED) (Mikolov et al., 2013) encode the categorical component  $\mathbf{c}^{(i)}$  using a representation trained directly within the model. Specifically, each categorical feature  $c_j^{(i)}$  (the  $j$ -th element of  $\mathbf{c}^{(i)}$ ) is assigned a trainable embedding vector of fixed dimensionality  $d_{\text{EMB}}$ . Hence, for each  $j \in \{1, \dots, D_{\text{cat}}\}$ , we have an embedding matrix

$$E_j \in \mathbb{R}^{K_j \times d_{\text{EMB}}}.$$



The embedding lookup operation retrieves the embedding vector for each  $c_j^{(i)}$ , and these vectors are then concatenated to form the full embedding for the categorical features:

$$\text{LEARNED}(\mathbf{c}^{(i)}) = \text{concat}(E_1[c_1^{(i)}], E_2[c_2^{(i)}], \dots, E_{D_{\text{cat}}}[c_{D_{\text{cat}}}^{(i)}]) \in \mathbb{R}^{D_{\text{cat}} \cdot d_{\text{EMB}}}. \quad (17)$$

To decode the learned embeddings back into categorical values, a nearest-neighbor approach is applied. For each embedding segment corresponding to a categorical feature, the pairwise distance between the embedding and the learned weights is computed, and the category with the minimum distance is selected:

$$\hat{c}_j^{(i)} = \arg \min_{k \in \{1, \dots, K_j\}} \|\tilde{\mathbf{c}}_j^{(i)} - E_j[k]\|, \quad (18)$$

where  $\tilde{\mathbf{c}}_j^{(i)} \in \mathbb{R}^{d_{\text{EMB}}}$  is the segment of the concatenated embedding corresponding to the  $j$ -th categorical feature, and  $E_j \in \mathbb{R}^{K_j \times d_{\text{EMB}}}$  is the embedding matrix for that feature.

**Analog Bits.** Analog Bits (I2B) (Chen et al., 2022) encode categorical features using a binary-based continuous representation. The encoding process involves two steps. We convert the categorical value to a real-valued binary representation where each category can be expressed using  $\lceil \log_2(K) \rceil$  binary bits based on the number of categories:

$$\text{I2B}(c_j^{(i)}, K_j^{(i)}) \in \mathbb{R}^{\lceil \log_2 K_j^{(i)} \rceil} \quad (19)$$

followed by a shift and scale formula:

$$\text{I2B}(c_j^{(i)}, K_j^{(i)}) \leftarrow (\text{I2B}(c_j^{(i)}, K_j^{(i)}) \cdot 2 - 1). \quad (20)$$

Thus, training and sampling of continuous-feature generative models (e.g., diffusion models) become computationally tractable. To decode, thresholding and rounding are applied to the generated continuous bits from the model to convert them back into binary form, which can be decoded trivially back into the original categorical values.

**Dictionary.** Dictionary encoding (DIC) represents categorical features using a continuous look-up embedding function. The encoding assigns equally spaced real-valued representations within a tunable specified range,  $[-1, 1]$ , to balance sparsity and separability. For categorical features with more categories, a wider range may be necessary to ensure proper distinction between values. The encoding process is defined as:

$$\text{DIC}(c_j^{(i)}, K_j^{(i)}) = -1 + \frac{2c_j^{(i)}}{K_j^{(i)} - 1}, \quad \text{DIC}(c_j^{(i)}, K_j^{(i)}) \in [-1, 1] \quad (21)$$

To decode, the nearest embedding is determined by comparing the encoded continuous value with all  $K_j^{(i)}$  possible embeddings, selecting the category with the smallest Euclidean distance.

## A.2. Architecture

**Flow Matching/DDPM Denoising Network.** The input layer projects the batch of tabular data input samples  $\mathbf{z}_t$ , each with dimension  $d_{in}$ , to the dimensionality  $d_t$  of our time step embeddings  $t_{emb}$  through a fully connected layer. This is so that we may leverage temporal information, which is appended to the result of the projection in the form of sinusoidal time step embeddings.

$$h_{in} = \text{FC}_{d_t}(\mathbf{z}_t) + t_{emb} \quad (22)$$

Subsequently, the output is passed through hidden layers  $h_1, h_2, h_3$ , and  $h_4$  which are fully connected networks used to learn the denoising direction or vector field. The output dimension of each layer is chosen as  $d_t, 2d_t, 2d_t$ , and  $d_t$  respectively. On top of the FC networks, each layer also consists of an activation function followed by dropout, as seen in the formulas below. This formulation is repeated for each hidden layer, at the end of which we obtain  $h_{out}$ . The exact activations, dropout, and other hyperparameters chosen are shown in Table 9.

$$h_1 = \text{Dropout}(\text{Activation}(\text{FC}(h_{in}))) \quad (23)$$

At last, the output layer transforms  $h_{out}$ , of dimension  $t_{emb}$  back to dimension  $d_{in}$  through a fully connected network, which now represents the score function  $\nabla_{\mathbf{z}_t} \log p_{\theta}(\mathbf{z}_t)$ .

$$\nabla_{\mathbf{z}_t} \log p_{\theta}(\mathbf{z}_t) = \text{FC}_{d_{in}}(h_{out}) \quad (24)$$

**Ordering of Categorical Feature.** Our CATCONVERTER representation induces an order on the categorical features. We assess how the order influences the results. By default, we assign a lexicographic ordering to the categorical features for simplicity purposes. We conducted an experiment to display the performance of lexicographic ordering vs. random ordering on the Adult and Beijing datasets. These datasets are chosen since they contain variables with a natural ordering such as “Education” (ordinal) and “Day” (periodic). As observed by the AUC disparity in Table 8, the order is an important factor in our CATCONVERTER representation. It also implicitly highlights that CATCONVERTER’s geometry aids in preserving the inherent semantics of ordinal and cyclical categorical features.

Table 8. AUC (classification) and RMSE (regression) scores of Machine Learning Efficiency. Higher scores indicate better performance.

METHODS	AUC $\uparrow$	RMSE $\downarrow$
	ADULT	BEIJING
TABGEN-DDPM (RANDOM)	0.776 $\pm$ .002	1.050 $\pm$ .006
TABGEN-DDPM (LEXICOGRAPHIC)	0.913 $\pm$ .002	0.508 $\pm$ .006
TABGEN-FLOW (RANDOM)	0.807 $\pm$ .003	1.041 $\pm$ .005
TABGEN-FLOW (LEXICOGRAPHIC)	0.912 $\pm$ .002	0.536 $\pm$ .006

**Out-of-index (OOI).** Out-of-index (OOI) can potentially occur due to the generative nature of DDPM and FM. For CATCONVERTER, OOI values are cast to the value of the 0-th index. Although casting ensures that all generated categorical values fall within the valid range, it introduces a bias. However, due to the generative capabilities of DDPM and FM, out-of-index values rarely occur as shown by our results.

### A.3. Hyperparameters

The hyperparameters selected for our model are shown below in Table 9.

Table 9. TABGEN Hyperparameters.

General		Flow Matching/DDPM Denoising MLP	
Hyperparameter	Value	Hyperparameter	Value
Training Iterations	100,000	Timestep embedding dimension $d_t$	1024
Flow Matching/DDPM Sampling Steps	50/1000	Activation	ReLU
Learning Rate	1e-4	Dropout	0.0
Weight Decay	5e-4	Hidden layer dimension $[h1, h2, h3, h4]$	[1024, 2048, 2048, 1024]
Batch Size	4096		
Optimizer	Adam		

## B. Experiments

### B.1. Datasets

Experiments were conducted with a total of 7 tabular datasets from the UCI Machine Learning Repository (Dua & Graff, 2017) with a (CC-BY 4.0) license. Classification tasks were performed on the Adult, Default, Shoppers, Stroke, and Diabetes datasets, while regression tasks were performed on the Beijing and News datasets. Each dataset was split into training, validation, and testing sets with a ratio of 8:1:1, except for the Adult dataset, whose official testing set was used and the remainder split into training and validation sets with an 8:1 ratio, and the Diabetes dataset, which was split into a ratio of 6:2:2. The resulting statistics of each dataset are shown in Table 10.

Table 10. Dataset Statistics. “# Num” and “# Cat” refer to the number of numerical and categorical columns.

Dataset	# Samples	# Num	# Cat	# Max Cat	# Train	# Validation	# Test	Task Type
<b>Adult</b>	48,842	6	9	42	28,943	3,618	16,281	Binary Classification
<b>Default</b>	30,000	14	11	11	24,000	3,000	3,000	Binary Classification
<b>Shoppers</b>	12,330	10	8	20	9,864	1,233	1,233	Binary Classification
<b>Beijing</b>	41,757	7	5	31	33,405	4,175	4,175	Regression
<b>News</b>	39,644	46	2	7	31,714	3,965	3,965	Regression
<b>Stroke</b>	4,909	3	8	5	3,927	490	490	Binary Classification
<b>Diabetes</b>	99,473	8	21	10	59,683	19,895	19,895	Multiclass Classification

## B.2. Baselines

TabUnite’s performance is evaluated in comparison to previous works in diffusion-based mixed-type tabular data generation. This includes STaSy (Kim et al., 2022), CoDi (Lee et al., 2023), TabDDPM (Kotelnikov et al., 2023), TabSYN (Zhang et al., 2023), and TabDiff (Shi et al., 2024b). The underlying architectures and implementation details of these models are presented in Table 11.

Table 11. Comparison of tabular data synthesis baselines.

Method	Model <sup>1</sup>	Type <sup>2</sup>	Categorical Encoding	Numerical Encoding	Additional Techniques
<b>STaSy</b>	Score-based Diffusion	U	One-Hot Encoding	Min-max scaler	Self-paced learning and fine-tuning.
<b>CoDi</b>	DDPM/Multinomial Diffusion	S	One-Hot Encoding	Min-max scaler	Model Inter-conditioning and Contrastive learning to learn dependencies between categorical and numerical data.
<b>TabDDPM</b>	DDPM/Multinomial Diffusion	S	One-Hot Encoding	Quantile Transformer	Concatenation of numerical and categorical features.
<b>TabSYN</b>	VAE + EDM	U	VAE-Learned	Quantile Transformer	Feature Tokenizer and Transformer encoder to learn cross-feature relationships with adaptive loss weighing to increase reconstruction performance.
<b>TabDiff</b>	EDM/Masked Diffusion	U	One-Hot Encoding	Quantile Transformer	Joint continuous-time diffusion process of numerical and categorical variables under learnable noise schedules, with a stochastic sampler to correct sampling errors.
<b>TABGEN-DDPM</b>	DDPM	U	CAT-CONVERTER	Quantile Transformer	Plug-and-play for Diffusion Models.
<b>TABGEN-Flow</b>	Flow Matching	U	CAT-CONVERTER	Quantile Transformer	Plug-and-play for Diffusion Models.

<sup>1</sup> The “Model” Column indicates the underlying architecture used for the model. Options include Denoising Diffusion Probabilistic Models or DDPMs (Ho et al., 2020b), Multinomial Diffusion (Hoogetboom et al., 2021), EDM, as introduced in (Karras et al., 2022).

<sup>2</sup> The “Type” column indicates the data integration approach used in the model. “U” denotes a unified data space where numerical and categorical data are combined after initial processing and fed collectively into the model. “S” represents a separated data space, where numerical and categorical data are processed and fed into distinct models.

### B.3. Benchmarks

We evaluate the generative performance on a broad suite of benchmarks. We analyze the capabilities in *downstream tasks* such as machine learning efficiency (MLE), where we determine the AUC score for classification tasks and RMSE for regression tasks of XGBoost (Chen & Guestrin, 2016) on the generated synthetic datasets. Next, we conduct experiments on *low-order statistics* where we perform column-wise density estimation (CDE) and pair-wise column correlation (PCC). Lastly, we examine the models’ quality on *high-order metrics* such as  $\alpha$ -precision and  $\beta$ -recall scores (Alaa et al., 2022). We add three additional benchmarks including a detection test metric, Classifier Two Sample Tests (C2ST) (SDMetrics, 2024), and privacy preservation metrics: the precision and recall of a membership inference attack (MIA) (Shokri et al., 2017). In this section, we expand on the concrete formulations behind our benchmarks including machine learning efficiency, low-order statistics, and high-order metrics. We also provide an overview on the detection and privacy metrics used in our experiments.

**Machine Learning Efficiency.** To evaluate the quality of our generated synthetic data, we use the data to train a classification/regression model, using XGBoost (Chen & Guestrin, 2016). This model is applied to the real test set. *AUC* (Area Under Curve) is used to evaluate the efficiency of our model in binary classification tasks. It measures the area under the Receiver Operating Characteristic (or ROC) curve, which plots the True Positive Rate against the False Positive Rate. AUC may take values in the range [0,1]. A higher AUC value suggests that our model achieves a better performance in binary classification tasks and vice versa.

$$\text{AUC} = \int_0^1 \text{TPR}(\text{FPR}) d(\text{FPR}) \quad (25)$$

*RMSE* (Root Mean Square Error) is used to evaluate the efficiency of our model in regression tasks. It measures the average magnitude of the deviations between predicted values ( $\hat{y}_i$ ) and actual values ( $y_i$ ). A smaller RMSE model indicates a better fit of the model to the data.

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (26)$$

**Low-Order Statistics.** *Column-wise Density Estimation* between numerical features is achieved with the Kolmogorov-Smirnov Test (KST). The Kolmogorov-Smirnov statistic is used to evaluate how much two underlying one-dimensional probability distributions differ, and is characterized by the below equation:

$$\text{KST} = \sup_x |F_1(x) - F_2(x)|, \quad (27)$$

where  $F_n(x)$ , the empirical distribution function of sample  $n$  is calculated by

$$F_n(x) = \frac{1}{n} \sum_{i=1}^n \mathbf{1}_{(-\infty, x]}(X_i) \quad (28)$$

*Column-wise Density Estimation* between two categorical features is determined by calculating the Total Variation Distance (TVD). This statistic captures the largest possible difference in the probability of any event under two different probability distributions. It is expressed as

$$\text{TVD} = \frac{1}{2} \sum_{x \in X} |P_1(x) - P_2(x)|, \quad (29)$$

where  $P_1(x)$  and  $P_2(x)$  are the probabilities (PMF) assigned to data point  $x$  by the two sample distributions respectively.

*Pair-wise Column Correlation* between two numerical features is computed using the Pearson Correlation Coefficient (PCC). It assigns a numerical value to represent the linear relationship between two columns, ranging from -1 (perfect negative linear correlation) to +1 (perfect positive linear correlation), with 0 indicating no linear correlation. It is computed as:

$$\rho(x, y) = \frac{\text{cov}(x, y)}{\sigma_x \sigma_y}, \quad (30)$$



To compare the Pearson Coefficients of our real and synthetic datasets, we quantify the dissimilarity in pair-wise column correlation between two samples

$$\text{Pearson Score} = \frac{1}{2} \mathbb{E}_{x,y} |\rho^1(x,y) - \rho^2(x,y)| \quad (31)$$

*Pair-wise Column Correlation* between two categorical features in a sample is characterized by a Contingency Table. This table is constructed by tabulating the frequencies at which specific combinations of the levels of two categorical variables work and recording them in a matrix format.

To quantify the dissimilarity of contingency matrices between two different samples, we use the Contingency Score.

$$\text{Contingency Score} = \frac{1}{2} \sum_{\alpha \in A} \sum_{\beta \in B} |P_{1,(\alpha,\beta)} - P_{2,(\alpha,\beta)}|, \quad (32)$$

where  $\alpha$  and  $\beta$  describe possible categorical values that can be taken in features  $A$  and  $B$ .  $P_{1,(\alpha,\beta)}$  and  $P_{2,(\alpha,\beta)}$  refer to the contingency tables representing the features  $\alpha$  and  $\beta$  in our two samples, which in this case corresponds to the real and synthetic datasets.

In order to obtain the column-wise density estimation and pair-wise correlation between a categorical and a numerical feature, we bin the numerical data into discrete categories before applying TVD and Contingency score respectively to obtain our low-order statistics.

We utilize the implementation of these experiments as provided by the SDMetrics library<sup>2</sup>.

**High-Order Statistics.** We utilize the implementations of High-Order Statistics as provided by the synthcity<sup>3</sup> library.  $\alpha$ -precision measures the overall fidelity of the generated data and is an extension of the classical machine learning quality metric of "precision". This formulation is based on the assumption that  $\alpha$  fraction of our real samples are characteristic of the original data distribution and the rest are outliers.  $\alpha$ -precision therefore quantifies the percentage of generated synthetic samples that match  $\alpha$  fraction of real samples.  $\beta$ -recall characterizes the diversity of our synthetic data and is similarly based on the quality metric of "recall".  $\beta$ -recall shares a similar assumption as  $\alpha$ -precision, except that we now assume that  $\beta$  fraction of our synthetic samples are characteristic of the distribution. Therefore, this measure obtains the fraction of the original data distribution represented by the  $\beta$  fraction of our generated samples (Alaa et al., 2022).

**Detection Metric: Classifier Two-Sample Test (C2ST).** The Classifier Two-Sample Test, a detection metric, assesses the ability to distinguish real data from synthetic data. This is done through a machine learning model that attempts to label whether a data point is synthetic or real. The score ranges from 0 to 1 where a score closer to 1 is superior, indicating that the machine learning model cannot concretely identify whether the data point is real or generated. We select logistic regression as our machine learning model, using the implementation provided by SDMetric (SDMetrics, 2024).

**Privacy Metric: Membership Inference Attacks (MIA).** Membership inference attacks evaluate the vulnerability of machine learning models to privacy leakage by determining whether a given instance was included in the training dataset (Shokri et al., 2017). The attacker often constructs a shadow model to mimic the target model's behavior and trains a binary classifier to distinguish membership status based on observed patterns.

These attacks are evaluated using precision, the fraction of inferred members that are true members, and recall, the fraction of true members correctly identified. When records are equally balanced between members and non-members, the ideal precision and recall are 0.5, indicating that the attack is no better than random guessing. Higher values suggest privacy leakage and reveal vulnerabilities in the model. Implementation of this metric is provided by SynthEval (Lautrup et al., 2024).

## C. Further Experimental Results

We perform experiments using several diffusion-based tabular generative model baselines, including STaSy (Kim et al., 2022), CoDi (Lee et al., 2023), TabDDPM (Kotelnikov et al., 2023), TabSYN (Zhang et al., 2023). We include TabDiff's reported results as the authors have not released code (Shi et al., 2024b).

<sup>2</sup><https://github.com/sdv-dev/SDMetrics>

<sup>3</sup><https://github.com/vanderschaarlab/synthcity>

We also incorporate several ablation experiments. We introduce TabFlow, an adaptation of TabDDPM (Kotelnikov et al., 2023) that models numerical and categorical tabular data with continuous flow matching (Lipman et al., 2022; Liu et al., 2022) and discrete flow matching (Tong et al., 2023), to examine the effect of unifying the data space. Experiments are also performed on various categorical representations on both diffusion and flow models. This includes one-hot, 1D-Learned Embedding, 2D-Learned Embedding, and Analog Bits (Chen et al., 2022), to demonstrate TABGEN’s effectiveness. We show that our proposed TABGEN framework achieves superior performance on the vast majority of metrics in Appendix C.1.

We evaluate AUC (classification), RMSE (regression), Column-Wise Density Estimation (CDE), Pair-Wise Column Correlation (PCC),  $\alpha$ -Precision,  $\beta$ -Recall scores, Classifier-Two Sample Test scores (C2ST), and Membership Inference Attacks Precision (MIA P.) and Recall (MIA R.) scores for our 7 datasets.  $\uparrow$  indicates that the higher the score, the better the performance;  $\downarrow$  indicates that the lower the score, the better the performance;  $\updownarrow$  indicates that an optimal score should be as close to 50% as possible.

The metrics and error bars shown in the tables in this section are derived from the mean and standard deviation of the experiments performed on 20 sampling iterations on the best-validated model.

### C.1. Additional Baseline Results

ADULT								
METHODS	AUC $\uparrow$	CDE $\uparrow$	PCC $\uparrow$	$\alpha$ $\uparrow$	$\beta$ $\uparrow$	C2ST $\uparrow$	MIA P. $\updownarrow$	MIA R. $\updownarrow$
STASY	0.906 $\pm$ 0.001	92.26 $\pm$ 0.04	89.15 $\pm$ 0.10	77.05 $\pm$ 0.29	33.54 $\pm$ 0.36	55.37	49.52 $\pm$ 0.50	24.51 $\pm$ 0.44
CoDI	0.871 $\pm$ 0.006	74.28 $\pm$ 0.08	77.38 $\pm$ 0.21	74.45 $\pm$ 0.35	8.74 $\pm$ 0.16	15.80	55.00 $\pm$ 7.26	0.05 $\pm$ 0.01
TABDDPM	0.910 $\pm$ 0.001	98.37 $\pm$ 0.08	96.69 $\pm$ 0.09	90.99 $\pm$ 0.35	<b>62.19<math>\pm</math>0.48</b>	97.55	51.30 $\pm$ 0.07	56.90 $\pm$ 0.29
TABSYN	0.906 $\pm$ 0.001	98.89 $\pm$ 0.03	97.56 $\pm$ 0.06	98.97 $\pm$ 0.26	47.68 $\pm$ 0.27	95.49	50.91 $\pm$ 0.16	42.91 $\pm$ 0.31
TABDIFF	0.912 $\pm$ 0.002	99.37 $\pm$ 0.05	98.51 $\pm$ 0.16	99.02 $\pm$ 0.20	51.64 $\pm$ 0.20	<b>99.50</b>	—	—
TABGEN-DDPM	<b>0.913<math>\pm</math>0.002</b>	<b>99.39<math>\pm</math>0.04</b>	<b>98.63<math>\pm</math>0.04</b>	<b>99.11<math>\pm</math>0.25</b>	52.04 $\pm$ 0.12	<b>99.50</b>	<b>50.44<math>\pm</math>0.83</b>	52.78 $\pm$ 0.21
TABGEN-FLOW	0.912 $\pm$ 0.002	98.63 $\pm$ 0.02	97.55 $\pm$ 0.23	98.21 $\pm$ 0.34	49.91 $\pm$ 0.28	95.48	50.65 $\pm$ 0.20	<b>50.51<math>\pm</math>0.21</b>
DEFAULT								
METHODS	AUC $\uparrow$	CDE $\uparrow$	PCC $\uparrow$	$\alpha$ $\uparrow$	$\beta$ $\uparrow$	C2ST $\uparrow$	MIA P. $\updownarrow$	MIA R. $\updownarrow$
STASY	0.752 $\pm$ 0.006	89.41 $\pm$ 0.03	92.64 $\pm$ 0.03	94.83 $\pm$ 0.15	40.23 $\pm$ 0.22	62.82	51.84 $\pm$ 0.69	30.37 $\pm$ 0.99
CoDI	0.525 $\pm$ 0.006	81.07 $\pm$ 0.08	86.25 $\pm$ 0.73	81.21 $\pm$ 0.11	19.75 $\pm$ 0.30	42.28	46.66 $\pm$ 2.54	3.41 $\pm$ 0.53
TABDDPM	0.761 $\pm$ 0.004	98.20 $\pm$ 0.05	97.16 $\pm$ 0.09	96.78 $\pm$ 0.30	<b>53.73<math>\pm</math>0.28</b>	97.12	51.34 $\pm$ 0.49	44.96 $\pm$ 0.59
TABSYN	0.755 $\pm$ 0.005	98.61 $\pm$ 0.08	<b>98.33<math>\pm</math>0.67</b>	98.49 $\pm$ 0.20	46.06 $\pm$ 0.37	95.83	50.80 $\pm$ 0.56	43.71 $\pm$ 0.89
TABDIFF	0.763 $\pm$ 0.005	98.76 $\pm$ 0.07	97.45 $\pm$ 0.75	98.49 $\pm$ 0.28	51.09 $\pm$ 0.25	97.74	—	—
TABGEN-DDPM	0.764 $\pm$ 0.005	<b>98.97<math>\pm</math>0.19</b>	96.74 $\pm$ 0.62	<b>98.66<math>\pm</math>0.24</b>	48.22 $\pm$ 0.48	<b>98.90</b>	50.07 $\pm$ 0.41	<b>48.96<math>\pm</math>0.41</b>
TABGEN-FLOW	<b>0.782<math>\pm</math>0.005</b>	97.45 $\pm$ 0.06	92.86 $\pm$ 1.75	96.50 $\pm$ 0.44	49.99 $\pm$ 0.23	89.36	<b>50.04<math>\pm</math>0.95</b>	47.07 $\pm$ 0.40
SHOPPERS								
METHODS	AUC $\uparrow$	CDE $\uparrow$	PCC $\uparrow$	$\alpha$ $\uparrow$	$\beta$ $\uparrow$	C2ST $\uparrow$	MIA P. $\updownarrow$	MIA R. $\updownarrow$
STASY	0.914 $\pm$ 0.005	82.53 $\pm$ 0.19	81.40 $\pm$ 0.27	68.18 $\pm$ 0.29	26.24 $\pm$ 0.60	25.82	51.23 $\pm$ 2.38	17.54 $\pm$ 0.19
CoDI	0.865 $\pm$ 0.006	67.27 $\pm$ 0.03	80.52 $\pm$ 0.12	90.52 $\pm$ 0.37	19.22 $\pm$ 0.27	19.04	59.67 $\pm$ 11.74	1.36 $\pm$ 0.45
TABDDPM	0.915 $\pm$ 0.004	97.58 $\pm$ 0.18	96.72 $\pm$ 0.22	90.85 $\pm$ 0.60	72.46 $\pm$ 0.46	83.49	51.24 $\pm$ 1.48	46.08 $\pm$ 1.57
TABSYN	0.918 $\pm$ 0.004	96.00 $\pm$ 0.12	95.18 $\pm$ 0.11	96.28 $\pm$ 0.24	45.79 $\pm$ 0.31	83.77	51.00 $\pm$ 1.08	42.14 $\pm$ 0.76
TABDIFF	0.921 $\pm$ 0.004	98.72 $\pm$ 0.09	<b>98.26<math>\pm</math>0.08</b>	<b>99.11<math>\pm</math>0.34</b>	49.75 $\pm$ 0.64	<b>98.43</b>	—	—
TABGEN-DDPM	<b>0.926<math>\pm</math>0.005</b>	<b>98.97<math>\pm</math>0.10</b>	97.62 $\pm$ 0.02	96.14 $\pm$ 0.19	53.68 $\pm$ 0.73	96.37	<b>49.86<math>\pm</math>0.98</b>	48.16 $\pm$ 0.90
TABGEN-FLOW	0.919 $\pm$ 0.005	97.74 $\pm$ 0.03	97.08 $\pm$ 0.07	95.85 $\pm$ 0.46	<b>55.92<math>\pm</math>0.37</b>	94.20	51.38 $\pm$ 1.66	<b>49.19<math>\pm</math>0.86</b>
DIABETES								
METHODS	AUC $\uparrow$	CDE $\uparrow$	PCC $\uparrow$	$\alpha$ $\uparrow$	$\beta$ $\uparrow$	C2ST $\uparrow$	MIA P. $\updownarrow$	MIA R. $\updownarrow$
STASY	0.629 $\pm$ 0.003	95.25 $\pm$ 0.03	93.41 $\pm$ 0.08	90.00 $\pm$ 0.17	39.62 $\pm$ 0.23	54.71	49.47 $\pm$ 0.22	29.75 $\pm$ 0.16
CoDI	0.469 $\pm$ 0.008	76.42 $\pm$ 0.02	78.07 $\pm$ 0.18	38.96 $\pm$ 0.16	6.39 $\pm$ 0.16	3.95	0.00 $\pm$ 0.00	0.00 $\pm$ 0.00
TABDDPM	0.634 $\pm$ 0.007	99.26 $\pm$ 0.01	98.71 $\pm$ 0.01	95.36 $\pm$ 0.32	<b>52.65<math>\pm</math>0.03</b>	92.18	50.40 $\pm$ 0.65	52.06 $\pm$ 0.11
TABSYN	0.634 $\pm$ 0.003	99.04 $\pm$ 0.01	98.32 $\pm$ 0.03	98.08 $\pm$ 0.14	45.08 $\pm$ 0.04	<b>93.38</b>	50.10 $\pm$ 0.34	44.42 $\pm$ 0.28
TABDIFF	—	—	—	—	—	—	—	—
TABGEN-DDPM	0.636 $\pm$ 0.003	<b>99.36<math>\pm</math>0.02</b>	<b>98.75<math>\pm</math>0.03</b>	97.19 $\pm$ 0.22	45.75 $\pm$ 0.49	92.65	<b>50.07<math>\pm</math>0.37</b>	<b>51.43<math>\pm</math>0.13</b>
TABGEN-FLOW	<b>0.637<math>\pm</math>0.002</b>	99.00 $\pm$ 0.02	98.46 $\pm$ 0.05	<b>99.08<math>\pm</math>0.13</b>	48.58 $\pm$ 0.17	90.41	50.24 $\pm$ 0.37	50.33 $\pm$ 0.13

**TABGEN: Training Tabular Diffusion Models with a Simple and Effective Continuous Representation**

STROKE								
METHODS	AUC $\uparrow$	CDE $\uparrow$	PCC $\uparrow$	$\alpha$ $\uparrow$	$\beta$ $\uparrow$	C2ST $\uparrow$	MIA P. $\downarrow$	MIA R. $\downarrow$
STaSY	0.833 $\pm$ 0.03	89.36 $\pm$ 0.13	84.99 $\pm$ 0.09	91.49 $\pm$ 0.33	39.92 $\pm$ 0.76	40.64	54.22 $\pm$ 1.14	34.63 $\pm$ 1.39
CoDI	0.798 $\pm$ 0.032	87.42 $\pm$ 0.17	80.65 $\pm$ 1.81	86.46 $\pm$ 0.53	28.59 $\pm$ 0.47	24.47	54.54 $\pm$ 1.60	27.32 $\pm$ 3.50
TABDDPM	0.808 $\pm$ 0.033	99.10 $\pm$ 0.05	97.09 $\pm$ 1.17	98.05 $\pm$ 0.14	<b>71.42<math>\pm</math>0.30</b>	<b>100.00</b>	53.45 $\pm$ 2.89	55.12 $\pm$ 0.81
TABSYN	0.845 $\pm$ 0.035	96.79 $\pm$ 0.08	95.18 $\pm$ 0.22	95.49 $\pm$ 0.41	48.85 $\pm$ 0.26	89.93	51.11 $\pm$ 1.54	47.97 $\pm$ 1.27
TABDIFF	—	—	—	—	—	—	—	—
TABGEN-DDPM	<b>0.869<math>\pm</math>0.027</b>	<b>99.14<math>\pm</math>0.20</b>	97.11 $\pm$ 0.60	<b>98.32<math>\pm</math>0.82</b>	57.17 $\pm$ 0.77	<b>100.00</b>	51.74 $\pm$ 1.85	<b>50.89<math>\pm</math>0.93</b>
TABGEN-FLOW	0.854 $\pm$ 0.028	98.42 $\pm$ 0.31	<b>97.37<math>\pm</math>2.12</b>	96.40 $\pm$ 0.71	63.91 $\pm$ 0.87	95.96	<b>50.66<math>\pm</math>1.77</b>	49.43 $\pm$ 1.16
BEIJING								
METHODS	RMSE $\downarrow$	CDE $\uparrow$	PCC $\uparrow$	$\alpha$ $\uparrow$	$\beta$ $\uparrow$	C2ST $\uparrow$	MIA P. $\downarrow$	MIA R. $\downarrow$
STaSY	0.656 $\pm$ 0.014	93.14 $\pm$ 0.07	90.63 $\pm$ 0.11	96.41 $\pm$ 0.10	51.35 $\pm$ 0.16	77.80	48.98 $\pm$ 0.78	34.06 $\pm$ 0.33
CoDI	0.818 $\pm$ 0.021	83.54 $\pm$ 0.04	90.35 $\pm$ 0.21	96.89 $\pm$ 0.14	53.16 $\pm$ 0.12	80.27	39.19 $\pm$ 6.35	0.40 $\pm$ 0.09
TABDDPM	0.592 $\pm$ 0.012	99.09 $\pm$ 0.02	96.71 $\pm$ 0.18	97.74 $\pm$ 0.06	<b>73.13<math>\pm</math>0.26</b>	95.13	50.43 $\pm$ 0.58	48.35 $\pm$ 0.79
TABSYN	0.555 $\pm$ 0.013	98.34 $\pm$ 0.01	96.85 $\pm$ 0.24	98.08 $\pm$ 0.27	55.68 $\pm$ 0.16	92.92	51.31 $\pm$ 0.42	46.53 $\pm$ 0.52
TABDIFF	0.555 $\pm$ 0.013	98.97 $\pm$ 0.05	<b>97.41<math>\pm</math>0.15</b>	98.06 $\pm$ 0.24	59.63 $\pm$ 0.23	97.81	—	—
TABGEN-DDPM	<b>0.508<math>\pm</math>0.006</b>	<b>99.11<math>\pm</math>0.03</b>	96.97 $\pm$ 0.20	<b>98.98<math>\pm</math>0.16</b>	64.08 $\pm$ 0.18	<b>98.16</b>	51.02 $\pm$ 0.42	<b>49.50<math>\pm</math>0.52</b>
TABGEN-FLOW	0.536 $\pm$ 0.006	98.28 $\pm$ 0.07	96.92 $\pm$ 0.21	98.16 $\pm$ 0.13	62.65 $\pm$ 0.12	92.26	<b>50.35<math>\pm</math>0.60</b>	49.14 $\pm$ 0.81
NEWS								
METHODS	RMSE $\downarrow$	CDE $\uparrow$	PCC $\uparrow$	$\alpha$ $\uparrow$	$\beta$ $\uparrow$	C2ST $\uparrow$	MIA P. $\downarrow$	MIA R. $\downarrow$
STaSY	0.871 $\pm$ 0.002	90.50 $\pm$ 0.10	96.59 $\pm$ 0.02	<b>97.95<math>\pm</math>0.14</b>	38.68 $\pm$ 0.30	51.72	49.68 $\pm$ 0.70	23.49 $\pm$ 0.67
CoDI	1.21 $\pm$ 0.005	70.82 $\pm$ 0.01	95.44 $\pm$ 0.05	86.03 $\pm$ 0.12	35.01 $\pm$ 0.13	9.35	40.00 $\pm$ 24.49	0.04 $\pm$ 0.02
TABDDPM	3.46 $\pm$ 1.25	94.79 $\pm$ 0.03	89.52 $\pm$ 0.10	90.94 $\pm$ 0.31	40.82 $\pm$ 0.42	0.02	50.72 $\pm$ 0.97	9.88 $\pm$ 0.52
TABSYN	0.866 $\pm$ 0.021	98.22 $\pm$ 0.04	98.53 $\pm$ 0.02	95.83 $\pm$ 0.33	43.97 $\pm$ 0.27	95.85	49.86 $\pm$ 0.75	34.42 $\pm$ 0.90
TABDIFF	0.866 $\pm$ 0.021	97.65 $\pm$ 0.03	98.72 $\pm$ 0.04	97.36 $\pm$ 0.17	42.10 $\pm$ 0.32	93.08	—	—
TABGEN-DDPM	<b>0.836<math>\pm</math>0.001</b>	<b>98.46<math>\pm</math>0.01</b>	<b>99.09<math>\pm</math>0.05</b>	95.35 $\pm$ 0.11	48.49 $\pm$ 0.12	<b>96.70</b>	<b>49.87<math>\pm</math>0.99</b>	<b>40.10<math>\pm</math>0.55</b>
TABGEN-FLOW	<b>0.814<math>\pm</math>0.002</b>	96.89 $\pm$ 0.03	98.34 $\pm$ 0.29	90.91 $\pm$ 0.25	<b>51.75<math>\pm</math>0.16</b>	88.13	50.90 $\pm$ 0.93	35.48 $\pm$ 0.78

## C.2. Additional Ablation Results

ADULT								
METHODS	AUC $\uparrow$	CDE $\uparrow$	PCC $\uparrow$	$\alpha$ $\uparrow$	$\beta$ $\uparrow$	C2ST $\uparrow$	MIA P. $\downarrow$	MIA R. $\downarrow$
ONEHOT-DDPM	0.476 $\pm$ 0.057	48.94 $\pm$ 0.12	38.05 $\pm$ 0.08	17.44 $\pm$ 0.21	0.70 $\pm$ 0.01	1.89	38.67 $\pm$ 19.02	0.03 $\pm$ 0.02
LEARNED1D-DDPM	0.611 $\pm$ 0.008	53.44 $\pm$ 3.39	27.97 $\pm$ 3.71	6.64 $\pm$ 0.05	0.00 $\pm$ 0.01	0.00	10.00 $\pm$ 10.00	0.00 $\pm$ 0.00
LEARNED2D-DDPM	0.793 $\pm$ 0.003	62.69 $\pm$ 1.16	38.53 $\pm$ 1.54	7.11 $\pm$ 0.21	0.02 $\pm$ 0.03	0.00	38.00 $\pm$ 5.61	0.03 $\pm$ 0.01
I2B-DDPM	0.911 $\pm$ 0.001	99.10 $\pm$ 0.07	97.55 $\pm$ 0.26	98.21 $\pm$ 0.18	47.44 $\pm$ 0.08	98.01	50.56 $\pm$ 0.15	50.68 $\pm$ 0.87
DIC-DDPM	0.912 $\pm$ 0.002	98.95 $\pm$ 0.03	97.65 $\pm$ 0.12	97.99 $\pm$ 0.54	51.09 $\pm$ 0.24	93.61	51.34 $\pm$ 0.20	51.62 $\pm$ 0.33
ONEHOT-FLOW	0.895 $\pm$ 0.003	90.66 $\pm$ 0.07	84.32 $\pm$ 0.10	88.35 $\pm$ 0.15	30.64 $\pm$ 0.13	38.88	51.08 $\pm$ 0.60	13.49 $\pm$ 0.36
LEARNED1D-FLOW	0.260 $\pm$ 0.014	60.00 $\pm$ 3.13	36.20 $\pm$ 3.61	6.82 $\pm$ 0.08	0.02 $\pm$ 0.03	0.00	24.67 $\pm$ 10.41	0.02 $\pm$ 0.01
LEARNED2D-FLOW	0.126 $\pm$ 0.012	62.37 $\pm$ 3.03	38.94 $\pm$ 4.20	6.64 $\pm$ 3.49	0.05 $\pm$ 0.04	0.00	28.19 $\pm$ 9.68	0.04 $\pm$ 0.01
I2B-FLOW	0.911 $\pm$ 0.001	98.23 $\pm$ 0.09	97.14 $\pm$ 0.32	<b>99.54<math>\pm</math>0.31</b>	48.87 $\pm$ 0.16	92.18	50.70 $\pm$ 0.31	<b>49.68<math>\pm</math>0.79</b>
DIC-FLOW	0.910 $\pm$ 0.002	98.10 $\pm$ 0.03	96.63 $\pm$ 0.03	99.64 $\pm$ 0.10	50.29 $\pm$ 0.12	90.70	50.55 $\pm$ 0.16	42.03 $\pm$ 0.42
TABFLOW	0.908 $\pm$ 0.002	96.21 $\pm$ 0.05	93.59 $\pm$ 0.05	86.76 $\pm$ 0.28	<b>53.15<math>\pm</math>0.14</b>	77.48	50.99 $\pm$ 0.38	43.90 $\pm$ 0.18
TABGEN-DDPM	<b>0.913<math>\pm</math>0.002</b>	<b>99.39<math>\pm</math>0.04</b>	<b>98.63<math>\pm</math>0.04</b>	99.11 $\pm$ 0.25	52.04 $\pm$ 0.12	<b>99.50</b>	<b>50.44<math>\pm</math>0.83</b>	52.78 $\pm$ 0.21
TABGEN-FLOW	0.912 $\pm$ 0.002	98.63 $\pm$ 0.02	97.55 $\pm$ 0.23	98.21 $\pm$ 0.34	49.91 $\pm$ 0.28	95.48	50.65 $\pm$ 0.20	50.51 $\pm$ 0.21
DEFAULT								
METHODS	AUC $\downarrow$	CDE $\uparrow$	PCC $\uparrow$	$\alpha$ $\uparrow$	$\beta$ $\uparrow$	C2ST $\uparrow$	MIA P. $\downarrow$	MIA R. $\downarrow$
ONEHOT-DDPM	0.557 $\pm$ 0.052	50.88 $\pm$ 0.10	50.88 $\pm$ 0.07	4.13 $\pm$ 0.05	0.15 $\pm$ 0.01	0.11	40.00 $\pm$ 24.49	0.08 $\pm$ 0.05
LEARNED1D-DDPM	0.575 $\pm$ 0.012	72.53 $\pm$ 1.92	51.95 $\pm$ 2.30	10.96 $\pm$ 2.03	0.00 $\pm$ 0.01	0.13	0.00 $\pm$ 0.00	0.00 $\pm$ 0.00
LEARNED2D-DDPM	0.290 $\pm$ 0.009	71.11 $\pm$ 0.32	50.79 $\pm$ 0.36	11.99 $\pm$ 2.54	0.01 $\pm$ 0.01	0.06	15.33 $\pm$ 11.62	0.11 $\pm$ 0.08
I2B-DDPM	0.762 $\pm$ 0.003	98.76 $\pm$ 0.06	<b>98.36<math>\pm</math>0.12</b>	98.20 $\pm$ 0.17	47.46 $\pm$ 0.37	98.34	50.95 $\pm$ 0.45	45.33 $\pm$ 0.77
DIC-DDPM	0.763 $\pm$ 0.007	98.33 $\pm$ 0.11	92.76 $\pm$ 0.43	97.20 $\pm$ 0.35	49.62 $\pm$ 0.37	96.88	51.60 $\pm$ 0.70	48.13 $\pm$ 0.75
ONEHOT-FLOW	0.759 $\pm$ 0.005	91.95 $\pm$ 0.05	88.04 $\pm$ 1.51	93.12 $\pm$ 0.31	30.50 $\pm$ 0.19	69.14	51.40 $\pm$ 1.23	12.32 $\pm$ 0.60
LEARNED1D-FLOW	0.438 $\pm$ 0.009	73.52 $\pm$ 0.82	53.01 $\pm$ 0.21	20.03 $\pm$ 5.43	0.00 $\pm$ 0.01	0.53	10.00 $\pm$ 10.00	0.05 $\pm$ 0.05
LEARNED2D-FLOW	0.709 $\pm$ 0.008	66.33 $\pm$ 4.86	44.11 $\pm$ 6.75	2.33 $\pm$ 12.11	0.01 $\pm$ 0.02	0.01	10.00 $\pm$ 10.00	0.03 $\pm$ 0.03
I2B-FLOW	0.763 $\pm$ 0.004	97.67 $\pm$ 0.13	94.65 $\pm$ 1.28	97.42 $\pm$ 0.57	49.15 $\pm$ 0.48	90.04	50.66 $\pm$ 0.33	43.09 $\pm$ 1.25
DIC-FLOW	0.759 $\pm$ 0.007	97.27 $\pm$ 0.03	92.26 $\pm$ 1.77	95.97 $\pm$ 0.27	51.29 $\pm$ 0.18	90.58	51.36 $\pm$ 0.62	45.95 $\pm$ 0.76
TABFLOW	0.742 $\pm$ 0.008	97.38 $\pm$ 0.03	95.01 $\pm$ 1.44	96.92 $\pm$ 0.12	<b>53.12<math>\pm</math>0.29</b>	85.89	<b>50.05<math>\pm</math>0.53</b>	44.11 $\pm$ 0.65
TABGEN-DDPM	0.764 $\pm$ 0.005	<b>98.97<math>\pm</math>0.19</b>	96.74 $\pm$ 0.62	<b>98.66<math>\pm</math>0.24</b>	48.22 $\pm$ 0.48	<b>98.90</b>	50.07 $\pm$ 0.41	<b>48.96<math>\pm</math>0.41</b>
TABGEN-FLOW	<b>0.782<math>\pm</math>0.005</b>	97.45 $\pm$ 0.06	92.86 $\pm$ 1.75	96.50 $\pm$ 0.44	49.99 $\pm$ 0.23	89.36	51.02 $\pm$ 0.95	47.07 $\pm$ 0.40

**TABGEN: Training Tabular Diffusion Models with a Simple and Effective Continuous Representation**

SHOPPERS								
METHODS	AUC $\uparrow$	CDE $\uparrow$	PCC $\uparrow$	$\alpha$ $\uparrow$	$\beta$ $\uparrow$	C2ST $\uparrow$	MIA P. $\downarrow$	MIA R. $\downarrow$
ONEHOT-DDPM	0.799 $\pm$ 0.126	90.37 $\pm$ 0.14	84.61 $\pm$ 0.10	90.67 $\pm$ 0.26	37.76 $\pm$ 0.57	54.59	51.00 $\pm$ 0.97	28.54 $\pm$ 1.49
LEARNED1D-DDPM	0.876 $\pm$ 0.028	78.94 $\pm$ 4.13	62.67 $\pm$ 6.40	21.94 $\pm$ 7.75	1.17 $\pm$ 0.81	1.36	59.09 $\pm$ 18.85	0.84 $\pm$ 0.26
LEARNED2D-DDPM	0.103 $\pm$ 0.011	70.97 $\pm$ 3.06	51.21 $\pm$ 4.37	8.43 $\pm$ 5.81	0.05 $\pm$ 0.09	0.09	0.00 $\pm$ 0.00	0.00 $\pm$ 0.00
I2B-DDPM	0.919 $\pm$ 0.005	98.67 $\pm$ 0.05	<b>98.00<math>\pm</math>0.03</b>	<b>97.69<math>\pm</math>0.63</b>	50.86 $\pm$ 0.11	96.28	51.77 $\pm$ 0.98	46.80 $\pm$ 1.91
DIC-DDPM	0.910 $\pm$ 0.005	97.83 $\pm$ 0.15	96.19 $\pm$ 0.09	95.33 $\pm$ 0.80	56.14 $\pm$ 0.85	91.09	50.68 $\pm$ 0.52	46.02 $\pm$ 1.20
ONEHOT-FLOW	0.910 $\pm$ 0.006	92.84 $\pm$ 0.08	91.50 $\pm$ 0.14	87.57 $\pm$ 0.44	48.77 $\pm$ 0.69	65.08	49.64 $\pm$ 1.69	34.56 $\pm$ 0.80
LEARNED1D-FLOW	0.134 $\pm$ 0.015	76.25 $\pm$ 2.16	60.05 $\pm$ 3.32	10.78 $\pm$ 7.79	0.10 $\pm$ 1.00	0.01	40.00 $\pm$ 24.49	0.13 $\pm$ 0.08
LEARNED2D-FLOW	0.868 $\pm$ 0.007	71.93 $\pm$ 0.94	53.29 $\pm$ 1.34	16.58 $\pm$ 1.72	0.28 $\pm$ 0.23	0.07	57.33 $\pm$ 20.50	0.65 $\pm$ 0.34
I2B-FLOW	0.910 $\pm$ 0.005	97.61 $\pm$ 0.11	97.20 $\pm$ 0.11	97.24 $\pm$ 0.66	54.88 $\pm$ 0.26	91.83	51.56 $\pm$ 1.01	45.11 $\pm$ 1.20
DIC-FLOW	0.903 $\pm$ 0.006	96.89 $\pm$ 0.14	95.78 $\pm$ 0.24	95.84 $\pm$ 0.38	52.39 $\pm$ 0.26	88.74	50.86 $\pm$ 0.71	52.53 $\pm$ 0.44
TABFLOW	0.914 $\pm$ 0.002	95.03 $\pm$ 0.04	92.87 $\pm$ 0.04	77.55 $\pm$ 0.19	<b>61.94<math>\pm</math>0.53</b>	73.74	51.62 $\pm$ 0.82	42.59 $\pm$ 0.74
TABGEN-DDPM	<b>0.926<math>\pm</math>0.005</b>	<b>98.97<math>\pm</math>0.10</b>	97.62 $\pm$ 0.02	96.14 $\pm$ 0.19	53.68 $\pm$ 0.73	<b>96.37</b>	<b>49.86<math>\pm</math>0.98</b>	48.16 $\pm$ 0.90
TABGEN-FLOW	0.919 $\pm$ 0.005	97.74 $\pm$ 0.03	97.08 $\pm$ 0.07	95.85 $\pm$ 0.46	55.92 $\pm$ 0.37	94.20	51.38 $\pm$ 1.66	<b>49.19<math>\pm</math>0.86</b>
DIABETES								
METHODS	AUC $\uparrow$	CDE $\uparrow$	PCC $\uparrow$	$\alpha$ $\uparrow$	$\beta$ $\uparrow$	C2ST $\uparrow$	MIA P. $\downarrow$	MIA R. $\downarrow$
ONEHOT-DDPM	0.484 $\pm$ 0.008	69.22 $\pm$ 0.04	50.14 $\pm$ 0.06	9.93 $\pm$ 0.16	2.34 $\pm$ 0.11	1.74	45.56 $\pm$ 0.03	0.36 $\pm$ 2.52
LEARNED1D-DDPM	0.464 $\pm$ 0.010	63.40 $\pm$ 4.59	39.94 $\pm$ 5.34	0.00 $\pm$ 0.00	0.00 $\pm$ 0.00	0.00	0.00 $\pm$ 0.00	0.00 $\pm$ 0.00
LEARNED2D-DDPM	0.516 $\pm$ 0.007	64.30 $\pm$ 1.31	41.49 $\pm$ 1.74	0.01 $\pm$ 0.01	0.00 $\pm$ 0.00	0.00	8.00 $\pm$ 0.00	0.00 $\pm$ 8.00
I2B-DDPM	0.632 $\pm$ 0.008	99.38 $\pm$ 0.01	98.84 $\pm$ 0.03	97.70 $\pm$ 0.13	46.83 $\pm$ 0.39	93.98	50.08 $\pm$ 0.39	51.63 $\pm$ 0.12
DIC-DDPM	0.636 $\pm$ 0.006	<b>99.50<math>\pm</math>0.02</b>	<b>99.12<math>\pm</math>0.01</b>	98.92 $\pm$ 0.13	48.34 $\pm$ 0.18	<b>95.03</b>	50.37 $\pm$ 0.25	54.48 $\pm$ 0.74
ONEHOT-FLOW	0.604 $\pm$ 0.005	96.65 $\pm$ 0.03	94.61 $\pm$ 1.99	97.43 $\pm$ 0.06	41.64 $\pm$ 0.16	55.44	49.49 $\pm$ 0.21	22.78 $\pm$ 0.26
LEARNED1D-FLOW	0.506 $\pm$ 0.007	53.27 $\pm$ 7.48	28.31 $\pm$ 8.51	0.00 $\pm$ 0.00	0.00 $\pm$ 0.00	0.00	0.00 $\pm$ 0.00	0.00 $\pm$ 0.00
LEARNED2D-FLOW	0.466 $\pm$ 0.008	68.39 $\pm$ 9.23	46.71 $\pm$ 10.95	0.00 $\pm$ 0.00	0.00 $\pm$ 0.00	0.00	0.00 $\pm$ 0.00	0.00 $\pm$ 0.00
I2B-FLOW	0.636 $\pm$ 0.003	98.92 $\pm$ 0.03	98.34 $\pm$ 0.05	98.45 $\pm$ 0.14	48.97 $\pm$ 0.29	89.28	49.86 $\pm$ 0.26	48.25 $\pm$ 0.13
DIC-FLOW	<b>0.637<math>\pm</math>0.007</b>	99.01 $\pm$ 0.03	98.54 $\pm$ 0.03	99.65 $\pm$ 0.10	48.98 $\pm$ 0.10	89.06	50.10 $\pm$ 0.12	48.01 $\pm$ 0.18
TABFLOW	0.618 $\pm$ 0.006	98.04 $\pm$ 0.02	96.82 $\pm$ 0.01	79.60 $\pm$ 0.09	<b>51.58<math>\pm</math>0.04</b>	73.35	50.19 $\pm$ 0.39	44.14 $\pm$ 0.15
TABGEN-DDPM	0.636 $\pm$ 0.003	99.36 $\pm$ 0.02	98.75 $\pm$ 0.03	97.19 $\pm$ 0.22	46.19 $\pm$ 0.49	92.65	<b>50.07<math>\pm</math>0.37</b>	51.43 $\pm$ 0.13
TABGEN-FLOW	<b>0.637<math>\pm</math>0.002</b>	99.00 $\pm$ 0.02	98.46 $\pm$ 0.05	<b>99.08<math>\pm</math>0.13</b>	48.58 $\pm$ 0.17	90.41	50.24 $\pm$ 0.37	<b>50.33<math>\pm</math>0.13</b>
STROKE								
METHODS	AUC $\uparrow$	CDE $\uparrow$	PCC $\uparrow$	$\alpha$ $\uparrow$	$\beta$ $\uparrow$	C2ST $\uparrow$	MIA P. $\downarrow$	MIA R. $\downarrow$
ONEHOT-DDPM	0.797 $\pm$ 0.033	98.74 $\pm$ 0.11	97.65 $\pm$ 0.10	96.78 $\pm$ 1.08	56.27 $\pm$ 0.22	98.62	53.08 $\pm$ 2.50	<b>49.76<math>\pm</math>1.24</b>
LEARNED1D-DDPM	0.743 $\pm$ 0.032	60.20 $\pm$ 11.13	35.86 $\pm$ 15.52	6.97 $\pm$ 42.82	0.26 $\pm$ 3.51	0.01	15.00 $\pm$ 0.20	0.33 $\pm$ 10.00
LEARNED2D-DDPM	0.850 $\pm$ 0.035	59.11 $\pm$ 14.59	33.77 $\pm$ 19.97	6.67 $\pm$ 36.39	0.87 $\pm$ 2.42	0.02	18.33 $\pm$ 0.33	0.49 $\pm$ 13.02
I2B-DDPM	0.852 $\pm$ 0.029	99.02 $\pm$ 0.18	95.12 $\pm$ 1.47	98.14 $\pm$ 0.16	64.11 $\pm$ 0.75	99.77	50.72 $\pm$ 2.05	49.11 $\pm$ 1.59
DIC-DDPM	0.824 $\pm$ 0.021	98.98 $\pm$ 0.09	<b>98.00<math>\pm</math>2.13</b>	98.25 $\pm$ 0.62	65.00 $\pm$ 0.88	99.39	52.97 $\pm$ 1.19	52.03 $\pm$ 1.76
ONEHOT-FLOW	0.812 $\pm$ 0.029	94.17 $\pm$ 0.08	90.45 $\pm$ 0.08	81.87 $\pm$ 0.55	49.23 $\pm$ 0.56	64.82	49.32 $\pm$ 1.47	39.51 $\pm$ 1.25
LEARNED1D-FLOW	0.142 $\pm$ 0.033	75.48 $\pm$ 6.55	56.60 $\pm$ 9.05	71.63 $\pm$ 35.72	0.45 $\pm$ 0.13	0.16	47.00 $\pm$ 0.55	1.14 $\pm$ 20.22
LEARNED2D-FLOW	0.180 $\pm$ 0.030	54.02 $\pm$ 13.09	28.95 $\pm$ 17.18	5.67 $\pm$ 24.97	0.21 $\pm$ 0.67	0.00	20.00 $\pm$ 0.16	0.16 $\pm$ 20.00
I2B-FLOW	0.797 $\pm$ 0.027	98.72 $\pm$ 0.11	94.65 $\pm$ 1.33	98.26 $\pm$ 0.27	65.23 $\pm$ 0.26	97.19	52.72 $\pm$ 2.26	52.03 $\pm$ 1.18
DIC-FLOW	0.807 $\pm$ 0.019	98.50 $\pm$ 0.23	92.71 $\pm$ 2.35	97.76 $\pm$ 0.64	65.77 $\pm$ 1.95	97.33	51.45 $\pm$ 1.29	48.78 $\pm$ 1.65
TABFLOW	0.868 $\pm$ 0.035	97.72 $\pm$ 0.01	96.00 $\pm$ 0.03	89.21 $\pm$ 0.82	<b>68.30<math>\pm</math>0.60</b>	89.19	51.80 $\pm$ 1.67	47.32 $\pm$ 1.95
TABGEN-DDPM	<b>0.869<math>\pm</math>0.027</b>	<b>99.14<math>\pm</math>0.20</b>	97.11 $\pm$ 0.60	<b>98.32<math>\pm</math>0.82</b>	57.17 $\pm$ 0.77	<b>100.00</b>	51.74 $\pm$ 1.85	50.89 $\pm$ 0.93
TABGEN-FLOW	0.854 $\pm$ 0.028	98.42 $\pm$ 0.31	97.37 $\pm$ 2.12	96.40 $\pm$ 0.71	63.91 $\pm$ 0.87	95.96	<b>50.66<math>\pm</math>1.77</b>	49.43 $\pm$ 1.16
BEIJING								
METHODS	RMSE $\downarrow$	CDE $\uparrow$	PCC $\uparrow$	$\alpha$ $\uparrow$	$\beta$ $\uparrow$	C2ST $\uparrow$	MIA P. $\downarrow$	MIA R. $\downarrow$
ONEHOT-DDPM	2.143 $\pm$ 0.339	74.21 $\pm$ 0.07	63.68 $\pm$ 0.05	48.88 $\pm$ 0.03	19.07 $\pm$ 0.13	19.68	50.28 $\pm$ 2.95	5.44 $\pm$ 0.50
LEARNED1D-DDPM	0.921 $\pm$ 0.006	79.49 $\pm$ 3.50	62.23 $\pm$ 5.61	26.94 $\pm$ 27.91	8.49 $\pm$ 5.95	13.51	45.45 $\pm$ 2.82	1.97 $\pm$ 0.13
LEARNED2D-DDPM	0.969 $\pm$ 0.005	81.89 $\pm$ 1.24	66.02 $\pm$ 2.37	79.49 $\pm$ 16.01	7.73 $\pm$ 1.53	55.83	49.20 $\pm$ 1.89	2.05 $\pm$ 0.10
I2B-DDPM	0.542 $\pm$ 0.008	98.66 $\pm$ 0.04	96.95 $\pm$ 0.21	97.92 $\pm$ 0.15	59.27 $\pm$ 0.14	94.28	50.64 $\pm$ 0.47	48.43 $\pm$ 0.89
DIC-DDPM	0.547 $\pm$ 0.007	98.83 $\pm$ 0.04	97.21 $\pm$ 0.16	98.97 $\pm$ 0.30	61.90 $\pm$ 0.12	<b>98.83</b>	51.32 $\pm$ 0.35	50.61 $\pm$ 0.66
ONEHOT-FLOW	0.765 $\pm$ 0.016	84.61 $\pm$ 0.02	67.28 $\pm$ 4.64	84.38 $\pm$ 0.61	20.32 $\pm$ 0.19	35.76	51.44 $\pm$ 1.49	8.12 $\pm$ 0.43
LEARNED1D-FLOW	0.806 $\pm$ 0.009	80.14 $\pm$ 1.60	64.19 $\pm$ 2.33	81.15 $\pm$ 3.68	13.11 $\pm$ 0.38	20.40	44.02 $\pm$ 3.58	1.13 $\pm$ 0.09
LEARNED2D-FLOW	0.787 $\pm$ 0.007	79.50 $\pm$ 0.85	63.04 $\pm$ 1.01	43.00 $\pm$ 4.90	7.55 $\pm$ 4.05	14.58	54.23 $\pm$ 1.72	2.01 $\pm$ 0.09
I2B-FLOW	0.543 $\pm$ 0.007	98.08 $\pm$ 0.04	96.87 $\pm$ 0.43	96.83 $\pm$ 0.12	60.58 $\pm$ 0.19	91.66	49.64 $\pm$ 0.47	45.77 $\pm$ 1.07
DIC-FLOW	0.561 $\pm$ 0.013	98.09 $\pm$ 0.03	96.37 $\pm$ 0.08	97.04 $\pm$ 0.22	60.78 $\pm$ 0.10	93.96	50.86 $\pm$ 0.71	52.53 $\pm$ 0.44
TABFLOW	0.574 $\pm$ 0.01	96.44 $\pm$ 0.06	93.71 $\pm$ 0.07	94.81 $\pm$ 0.42	59.47 $\pm$ 0.28	87.23	50.54 $\pm$ 0.35	42.20 $\pm$ 0.61
TABGEN-DDPM	<b>0.508<math>\pm</math>0.006</b>	<b>99.11<math>\pm</math>0.03</b>	<b>97.97<math>\pm</math>0.20</b>	<b>98.98<math>\pm</math>0.16</b>	<b>64.08<math>\pm</math>0.18</b>	98.16	51.02 $\pm$ 0.42	<b>49.50<math>\pm</math>0.52</b>
TABGEN-FLOW	0.536 $\pm$ 0.006	98.28 $\pm$ 0.07	96.92 $\pm$ 0.21	98.16 $\pm$ 0.13	62.65 $\pm$ 0.12	92.26	<b>50.35<math>\pm</math>0.60</b>	49.14 $\pm$ 0.81



METHODS	NEWS							
	RMSE ↓	CDE ↑	PCC ↑	$\alpha$ ↑	$\beta$ ↑	C2ST ↑	MIA P. ↓	MIA R. ↓
ONEHOT-DDPM	0.840 $\pm$ 0.02	98.11 $\pm$ 0.06	92.78 $\pm$ 0.08	96.33 $\pm$ 0.24	46.87 $\pm$ 0.28	95.80	50.25 $\pm$ 0.46	32.42 $\pm$ 0.80
LEARNED1D-DDPM	0.858 $\pm$ 0.01	96.45 $\pm$ 0.14	95.00 $\pm$ 0.32	90.48 $\pm$ 9.11	19.54 $\pm$ 4.84	21.58	51.33 $\pm$ 0.96	17.60 $\pm$ 1.02
LEARNED2D-DDPM	0.857 $\pm$ 0.023	96.06 $\pm$ 0.49	94.49 $\pm$ 0.84	88.83 $\pm$ 4.52	5.11 $\pm$ 5.24	20.41	50.85 $\pm$ 2.97	3.83 $\pm$ 0.39
I2B-DDPM	0.844 $\pm$ 0.013	98.41 $\pm$ 0.02	98.57 $\pm$ 0.18	95.02 $\pm$ 0.10	48.22 $\pm$ 0.18	96.07	50.88 $\pm$ 0.57	39.82 $\pm$ 0.93
DIC-DDPM	0.866 $\pm$ 0.019	98.22 $\pm$ 0.06	98.50 $\pm$ 0.36	97.08 $\pm$ 0.25	47.88 $\pm$ 0.10	95.75	49.54 $\pm$ 0.57	39.80 $\pm$ 0.35
ONEHOT-FLOW	0.850 $\pm$ 0.017	96.27 $\pm$ 0.05	98.11 $\pm$ 0.02	<b>97.78<math>\pm</math>0.13</b>	43.06 $\pm$ 0.62	84.56	50.45 $\pm$ 2.02	14.84 $\pm$ 0.65
LEARNED1D-FLOW	0.873 $\pm$ 0.007	95.07 $\pm$ 0.14	95.07 $\pm$ 0.12	89.16 $\pm$ 5.89	18.17 $\pm$ 6.18	79.17	49.85 $\pm$ 1.69	10.28 $\pm$ 0.47
LEARNED2D-FLOW	0.866 $\pm$ 0.005	94.85 $\pm$ 0.38	94.99 $\pm$ 0.89	80.66 $\pm$ 15.23	14.63 $\pm$ 4.14	78.96	49.59 $\pm$ 1.54	14.19 $\pm$ 0.69
I2B-FLOW	0.847 $\pm$ 0.014	96.64 $\pm$ 0.05	98.38 $\pm$ 0.34	88.39 $\pm$ 0.11	<b>51.85<math>\pm</math>0.24</b>	89.47	51.12 $\pm$ 0.71	36.35 $\pm$ 0.53
DIC-FLOW	0.853 $\pm$ 0.014	96.58 $\pm$ 0.04	97.49 $\pm$ 0.34	92.28 $\pm$ 0.25	50.79 $\pm$ 0.29	88.30	50.43 $\pm$ 0.63	37.56 $\pm$ 0.77
TABFLOW	0.850 $\pm$ 0.017	96.51 $\pm$ 0.08	97.93 $\pm$ 0.02	92.68 $\pm$ 0.31	50.03 $\pm$ 0.26	87.33	51.01 $\pm$ 0.24	28.00 $\pm$ 0.89
TABGEN-DDPM	0.836 $\pm$ 0.001	<b>98.46<math>\pm</math>0.01</b>	<b>99.09<math>\pm</math>0.05</b>	95.35 $\pm$ 0.11	48.49 $\pm$ 0.12	<b>96.70</b>	<b>49.87<math>\pm</math>0.99</b>	<b>40.10<math>\pm</math>0.55</b>
TABGEN-FLOW	<b>0.814<math>\pm</math>0.002</b>	96.89 $\pm$ 0.03	98.34 $\pm$ 0.29	90.91 $\pm$ 0.25	51.75 $\pm$ 0.16	88.13	50.90 $\pm$ 0.93	35.48 $\pm$ 0.78