LEARNING DIFFUSION BRIDGES ON CONSTRAINED DOMAINS

Anonymous authors
Paper under double-blind review

ABSTRACT

Diffusion models have achieved promising results on generative learning recently. However, because diffusion processes are most naturally applied on the unconstrained Euclidean space $\mathbb{R}^d$, key challenges arise for developing diffusion based models for learning data on constrained and structured domains. We present a simple and unified framework to achieve this that can be easily adopted to various types of domains, including product spaces of any type (be it bounded/unbounded, continuous/discrete, categorical/ordinal, or their mix). In our model, the diffusion process is driven by a drift force that is a sum of two terms: one singular force designed by *Doob's h-transform* that ensures all outcomes of the process to belong to Ω, and one non-singular neural force field that is trained to make sure the outcome follows the data distribution statistically. Experiments show that our methods perform superbly on generating tabular data, images, semantic segments and 3D point clouds.

1 INTRODUCTION

Diffusion-based deep generative models, notably score matching with Langevin dynamics (SMLD) (Song & Ermon, 2019, 2020), denoising diffusion probabilistic models (DDPM) (Ho et al., 2020), and their variants (e.g., Song et al., 2020b,a; Kong & Ping, 2021; Song et al., 2021; Nichol & Dhariwal, 2021), have shown to achieve new state of the art results for image synthesis (Dhariwal & Nichol, 2021; Ramesh et al., 2022; Ho et al., 2022; Liu et al., 2021), audio synthesis (Chen et al., 2020; Kong et al., 2020), point cloud synthesis (Luo & Hu, 2021a,b; Zhou et al., 2021), and many other AI tasks. These methods train a deep neural network to drive as drift force a diffusion process to generate data, and are shown to outperform competitors, mainly GANs and VAEs, on stability and sample diversity (Xiao et al., 2021; Ho et al., 2020; Song et al., 2020b).

However, due to the continuous nature of diffusion processes, the standard approaches are restricted to generating unconstrained continuous data in $\mathbb{R}^d$. For generating data constrained on special structured domains, such as discrete, bound data or mixes of them, special techniques , e.g., dequantization (Uria et al., 2013; Ho et al., 2019) and multinomial diffusion (Hoogeboom et al., 2021; Austin et al., 2021), need to be developed case by case and the results still tend to be unsatisfying despite promising recent advances (Hoogeboom et al., 2021; Austin et al., 2021).

This work proposes a simple and unified framework for learning diffusion models on general constrained domains Ω embedded in the Euclidean space $\mathbb{R}^d$. The idea is to learn a continuous $\mathbb{R}^d$-valued diffusion process Z_t on time interval $t \in [0, T]$, with a carefully designed force field, such that the final state Z_T guarantees to 1) fall into the desirable domain Ω, and 2) follows the data distribution asymptotically. We achieve both steps by leveraging a key tool in stochastic calculus called *Doob's h-transform* (Doob, 1984), which provides formula for deriving the diffusion processes whose final states are guaranteed to fall into a specific set or equal a specific value.

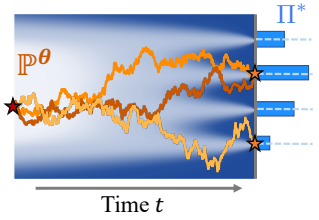


Figure 1: A 1D Ω-Bridges for discrete $\Omega = \{1, 2, 3, 4\}$.

Our simple procedure can be applied to any domain Ω once a properly defined summation (for discrete sets) or integration (for continuous domains) can be evaluated. To give a quick overview on the practical intuition without invoking the mathematical theory, we show in Algorithm 1 an simple

Algorithm 1 Learning Diffusion Models on Constrained Domains (a Simple Example)

Input: A dataset $\mathcal{D} \coloneqq \{x^{(i)}\}$ drawn from distribution Π^* on a domain $\Omega = \{e_1, e_2, \dots, e_K\}$.

Goal: Learn a diffusion model that terminates at time T to generate samples from Π^*.

Learning: Solve the optimization below with stochastic gradient descent (or other optimizers)

$$\theta^* = \arg\min_\theta \int_0^T \mathbb{E}_{x\sim\mathcal{D}} \left[\left\| f^\theta(Z_t, t) - \nabla_{Z_t} \log \omega^\Omega(x \mid Z_t, t) \right\|^2 \right] \mathrm{d}t, \tag{1}$$

where

$$\omega^\Omega(x \mid z, t) = \frac{\exp\left(-\dfrac{\|z-x\|^2}{2(T-t)}\right)}{\displaystyle\sum_{e\in\Omega} \exp\left(-\dfrac{\|z-e\|^2}{2(T-t)}\right)}, \qquad Z_t = \frac{t}{T}x + (1-\frac{t}{T})x_0 + \sqrt{\frac{t(T-t)}{T}}\xi, \tag{2}$$

with x drawn from the dataset $\mathcal{D}$, $\xi \sim \mathcal{N}(0, I)$, and x_0 any initial point.

Sampling: Generate sample Z_T from

$$\mathrm{d}Z_t = \left[f^{\theta^*}(Z_t, t) + \nabla_{Z_t} \log \sum_{e\in\Omega} \exp\left(-\frac{\|Z_t - e\|^2}{2(T-t)}\right) \right] \mathrm{d}t + \mathrm{d}W_t, \quad Z_0 = x_0.$$

Remark When the domain Ω is a manifold (e.g., line or surface) in $\mathbb{R}^d$, simply replace the sum $\sum_{e\in\Omega}$ with the corresponding line or surface (in general Hausdorff) integration $\int_\Omega$ on Ω.

instance of the framework when the domain is a discrete set $\Omega = \{e_1, \dots, e_K\}$. The idea is to set up the diffusion model to have a form of

$$\mathrm{d}Z_t = \left[f^\theta(Z_t, t) + \nabla_{Z_t} \psi^\Omega(Z_t, t) \right] \mathrm{d}t + \mathrm{d}W_t, \quad \psi^\Omega(z, t) \coloneqq \log \sum_{e\in\Omega} \exp\left(-\frac{\|z-e\|^2}{2(T-t)}\right), \tag{3}$$

where the drift is a sum of a non-singular (e.g., bounded) term $f^\theta(z, t)$ which is a trainable neural force field with parameter θ, and a singular term $\nabla_z \psi^\Omega(z, t)$, which drives Z_t towards set Ω as a gradient ascent on $\psi^\Omega(z, t)$. The $\psi^\Omega(z, t)$ measures the closeness of z to set Ω, as the log-likelihood of a Gaussian mixture model (GMM) centered on the elements in Ω with variance $T - t$. When t approaches to the terminal time T, the variance $T - t$ of the GMM goes to zero, and the magnitude of $\nabla_z \psi^\Omega(z, t)$ grows to infinity, hence ensuring that Z_T must belong to Ω. In particular, note that

$$\nabla_z \psi^\Omega(z, t) = \sum_{e\in\Omega} \omega^\Omega(e \mid z, t) \frac{e - z}{T - t}, \qquad \omega^\Omega(e \mid z, t) \propto \frac{\exp\left(-\frac{\|z-e\|^2}{2(T-t)}\right)}{\exp(\psi^\Omega(z, t))},$$

which increases with an $O(1/(T - t))$ rate as $t \to T$; here $\omega^\Omega(e \mid z, t)$ is the softmax probability measuring the relative closeness of z to the elements e in Ω (see also Eq (2)).

By Girsanov theorem (e.g., Oksendal, 2013), once f^θ is non-singular in the sense that $\int_0^T \mathbb{E}[\|f^\theta(Z_t, t)\|^2]\mathrm{d}t < +\infty$ (which is satisfied when f^θ is bounded), the diffusion model in (3) guarantees to yield a final state Z_T that belongs to Ω, and hence provides a flexible model family on Ω. Moreover, as shown in Eq 1 in Algorithm 1, the neural field f^θ can be simply trained to approximate $\nabla \log \omega^\Omega(e \mid z, t)$ with e plugged as the data point that we expect to achieve when starting from z at time t. Intuitively, such fitted f^θ increases the relative probability of the observed data points and hence allows us to fit the data distribution. Empirically, diffusion models learned through Ω-bridge achieves favorable results in generating mixed discrete/continuous tabular data, point clouds on grids, categorical semantic segments and discrete CIFAR10 images.

Outline The rest of the paper is organized as follows. Section 2 introduces h-transform, which allows us to derive bridge processes that are guaranteed to enter specific sets at the terminal time,

and Section 2.3 specifies the parametric diffusion models for Ω-bridges. Then, with the learnable diffusion models, Section 3 introduces the general learning framework along with the loss function.

2 BACKGROUND: DIFFUSION PROCESSES AND h-TRANSFORM

A diffusion process $\boldsymbol{Z} = \{Z_t \colon t \in [0, T]\}$ on $\mathbb{R}^d$ follows a stochastic differential equation of form

$$\mathbb{Q}: \qquad \mathrm{d}Z_t = b(Z_t, t)\mathrm{d}t + \sigma(Z_t, t)\mathrm{d}W_t, \tag{4}$$

where W_t is a Wiener process, and $\sigma \colon [0, T] \times \mathbb{R}^d \to \mathbb{R}$ is a positive diffusion coefficient, and $b \in [0, T] \times \mathbb{R}^d \to \mathbb{R}^d$ is a drift function. We use $\mathbb{Q}$ (or $\mathbb{P}$) to denote the path measure of stochastic processes $\boldsymbol{Z}$, which are distributions on the probability measures on the space of continuous paths. Let $\mathbb{Q}_t$ be the marginal distribution of Z_t at time t under $\mathbb{Q}$.

Our framework heavily relies on the *bridge processes*, special stochastic processes that guarantee to achieve a deterministic value or fall into a given set at the final state T.

☞ *For a set $\Omega \subseteq \mathbb{R}^d$, a process Z in $\mathbb{R}^d$ with law $\mathbb{Q}$ is called an Ω-bridge if $\mathbb{Q}(Z_T \in \Omega) = 1$.*

One natural approach to constructing bridge processes is to derive the conditioned process of a general unconstrained process given that the desirable bridge constraint happens. Assume that $\mathbb{Q}$ is the law of a general unconstrained diffusion process of form (4). Specifically, denote by $\mathbb{Q}^\Omega(\cdot) = \mathbb{Q}(\cdot \mid Z_T \in \Omega)$ the conditioned distribution given that the event of $Z_T \in \Omega$ happens. Then $\mathbb{Q}^\Omega$ is guaranteed to be an Ω-bridge by definition. Importantly, a remarkable result from Doob (Doob, 1984), now known as h-transform, shows that $\mathbb{Q}^\Omega$ is the law of a diffusion process with a properly modified drift term. Below, we introduce this results, first for the case x-bridge when $\Omega = \{x\}$ includes a single point, and then for more general sets Ω. For simplicity, we only state the formula from h-transform that are useful for us without proofs. Interested readers are referred to (e.g., Oksendal, 2013; Rogers & Williams, 2000) for more background on h-transform.

2.1 x-BRIDGES

Let us first consider the x-bridges $\mathbb{Q}^x(\cdot) \coloneqq \mathbb{Q}(\cdot \mid Z_T = x)$, the process $\mathbb{Q}$ pinned at a deterministic terminal point $Z_T = x$. By the result from h-transform (see e.g., Oksendal (2013)), the conditioned process $\mathbb{Q}^x(\cdot) \coloneqq \mathbb{Q}(\cdot \mid Z_T = x)$, if it exists, can be shown to be the law of

$$\mathrm{d}Z_t = \big(b(Z_t, t) + \sigma^2(Z_t, t)\nabla_z \log q_{T|t}(x \mid Z_t)\big)\,\mathrm{d}t + \sigma(Z_t, t)\mathrm{d}W_t, \tag{5}$$

where $q_{T|t}(x|z)$ is the density function of the transition probability

$$\mathbb{Q}_{T|t}(\mathrm{d}x \mid z) = \mathbb{Q}(Z_T \in \mathrm{d}x \mid Z_t = z),$$

assuming it exists. Here $\mathrm{d}x$ denotes an infinitesimal volume around x. Compared with the diffusion process (4) of $\mathbb{Q}$, the main difference is that the conditioned process has an additional drift force $\sigma^2(z, t)\nabla_z \log q_{T|t}(x|z)$ which plays the role of steering Z_t towards the target $Z_T = x$; this is a singular force whose magnitude increases to infinity as $t \to T$, because $q_{T|t}(\cdot \mid z)$ is a delta measure centered at z when $t = T$.

In addition, by Bayes rule, the distribution of the initial state Z_0 should be given by

$$Z_0 \sim \mathbb{Q}_{0|T}(\cdot \mid x), \qquad \mathbb{Q}_{0|T}(\mathrm{d}z|x) \propto \mathbb{Q}_0(\mathrm{d}z)q_{T|0}(x|z). \tag{6}$$

Example 2.1. *If $\mathbb{Q}$ is the law of $\mathrm{d}Z_t = \sigma_t \mathrm{d}W_t$, we have $\mathbb{Q}_{T|t}(\cdot|z) = \mathcal{N}(z, \beta_T - \beta_t)$, where $\beta_t = \int_0^t \sigma_s^2 \mathrm{d}s$. Hence, following the formula in (5), $\mathbb{Q}^x \coloneqq \mathbb{Q}(\cdot|Z_T = x)$ is the law of*

$$\mathrm{d}Z_t = \sigma_t^2 \frac{x - Z_t}{\beta_T - \beta_t}\mathrm{d}t + \sigma_t \mathrm{d}W_t, \tag{7}$$

and $Z_0 \sim \mathbb{Q}_{0|T}(\mathrm{d}z) \propto \mathbb{Q}_0(\mathrm{d}z)\phi(x \mid z, \beta_T - \beta_t)$, and $\phi(\cdot|\mu, \sigma^2)$ is the density function of $\mathcal{N}(\mu, \sigma^2)$. The process in (7) is known as a (time-scaled) Brownian bridge. Note that the drift in (7) grows to infinity in magnitude with a rate of $O(1/(\beta_T - \beta_t))$ as $t \to T$, which ensures that $Z_t = x$ with probability one.

Arbitrary initialization To make (5) the conditioned process of (4), the initial distribution must follow the Bayes rule in (6). However, thanks to the singular force $\nabla_z \log q_{T|t}(x|z)$, the process (5) can guarantee $Z_t^x = x$ from an arbitrary initialization once the process is well defined. When the initialization is different from (6), the process in (5) is no longer the conditioned process of (4), but it remains to be an x-bridge in that $Z_T = x$ is still guaranteed. To see why this is the case, assume that $\mathbb{Q}$ is initialized from a deterministic point $Z_0 = x_0$. Then we would still have $Z_0 = x_0$ when conditioned on $Z_T \in \Omega$ by Bayes rule. This suggests that (5) starting from any deterministic initialization is the condition process of $\mathbb{Q}$ with the same deterministic initialization, and is hence an x-bridge. As a result, (5) from any stochastic initialization is also an x-bridge because it can be viewed as the mixture of the processes with different deterministic initialization, all of which are x-bridges. See Appendix A.2 for a detailed analysis, in which it is shown that (5) with an arbitrary initialization can be viewed as the conditioned process of a special class of non-Markov processes called *reciprocal process.*

2.2 Ω-Bridges

More generally, for the law $\mathbb{Q}$ of (4) and a set $\Omega \in \mathbb{R}^d$, the Ω-bridge $\mathbb{Q}^\Omega := \mathbb{Q}(\cdot \mid Z_T \in \Omega)$ follows

$$\mathbb{Q}^\Omega : \qquad \mathrm{d}Z_t = \eta^\Omega(Z_t, t)\mathrm{d}t + \sigma(Z_t, t)\mathrm{d}W_t, \tag{8}$$

with

$$\eta^\Omega(z,t) = b(z,t) + \sigma^2(z,t)\mathbb{E}_{x\sim\mathbb{Q}_{T|t,z,\Omega}}[\nabla_z \log q_{T|t}(x \mid z)], \qquad Z_0 \sim \mathbb{Q}_{0|T}(\cdot \mid Z_T \in \Omega),$$

where drift force η^Ω is similar to that of the x-bridge in (5), except that x is now randomly drawn from an "Ω-truncated" transition probability:

$$\mathbb{Q}_{T|t,z,\Omega}(\mathrm{d}x \mid z) := \mathbb{Q}(Z_T = \mathrm{d}x \mid Z_t = z, Z_T \in \Omega),$$

which is the transition probability from Z_t to Z_T, conditioned on that $Z_T \in \Omega$. In practice, its form can be derived using Bayes rule.

Example 2.2. *Assume $\mathbb{Q}$ follows $\mathrm{d}Z_t = \sigma_t \mathrm{d}W_t$. Then $\mathbb{Q}^\Omega$ yields the following Ω-bridge:*

$$\mathrm{d}Z_t = \eta^\Omega(Z_t,t)\mathrm{d}t + \sigma_t \mathrm{d}W_t, \qquad \eta^\Omega(z,t) = \sigma_t^2 \mathbb{E}_{x\sim\mathcal{N}_\Omega(z,\beta_T-\beta_t)}\left[\frac{x-z}{\beta_T-\beta_t}\right], \tag{9}$$

where $\mathcal{N}_\Omega(z,\sigma^2) = \mathrm{Law}(Z \mid Z \in \Omega)$ when $Z \sim \mathcal{N}(\mu,\sigma^2)$, which is an Ω-truncated Gaussian distribution $\mathcal{N}(\mu,\sigma^2)$, whose density function is $\phi_\Omega(x) \propto \mathbb{I}(x \in \Omega)\phi(x|\mu,\sigma)$ with $\phi(x|\mu,\sigma)$ the density function of $\mathcal{N}(\mu,\sigma^2)$.

Note that it is tractable to calculate η^Ω once we can evaluate the expectation of $\mathcal{N}_\Omega(z,\beta_T-\beta_t)$. A general case is when $\Omega = I_1 \times \cdots I_d$, for which the expectation reduces to one dimensional Gaussian integrals. See Appendix A.3 and A.4 for details and examples of η^Ω.

As in the x-bridge, we can set the initialization to be any distribution supported on the set of points that can reach Ω following $\mathbb{Q}$ (precisely, points z_0 that satisfy $\Omega \cap \mathrm{supp}(\mathbb{Q}_T(\cdot|Z_0 = z_0)) \neq \emptyset$) using the mixture of initialization argument.

2.3 A Parametric Family of Ω-Bridges

The formula in (8) only provides a fixed process for a given $\mathbb{Q}$. For the purpose of learning generative models, however, we need a rich family of Ω-bridges within which we can search for a best one to fit with the data distribution. It turns out we can achieve this by simply adding an extra non-singular drift force, which can be a trainable neural network, on top of the Ω-bridge in (8). Specifically, we construct the following parametric diffusion model $\mathbb{P}^\theta$:

$$\mathbb{P}^\theta : \qquad \mathrm{d}Z_t = (\sigma(Z_t,t)f^\theta(Z_t,t) + \eta^\Omega(Z_t,t))\mathrm{d}t + \sigma(Z_t,t)\mathrm{d}W_t, \quad Z_0 \sim \mathbb{P}_0^\theta, \tag{10}$$

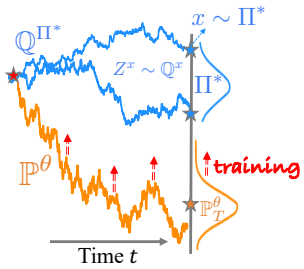
where $f^\theta(z,t)$ is a neural network with input (z,t) and parameter θ, which will be trained based on the empirical observations. Adding the neural drift $\sigma(Z_t,t)f^\theta(Z_t,t)$ term does not break the Ω-bridge condition, once it satisfies a very mild regularization condition:

Proposition 2.3. *For any $\mathbb{Q}^\Omega$ following $\mathrm{d}Z_t = \eta^\Omega(Z_t,t)\mathrm{d}t + \mathrm{d}W_t$ that is an Ω-bridge, the $\mathbb{P}^\theta$ in (10) is also an Ω-bridge if $\mathbb{E}_{Z\sim\mathbb{Q}^\Omega}[\int_0^T \left\|f^\theta(Z_t,t)\right\|_2^2 \mathrm{d}t] < +\infty$ and $\mathcal{KL}(\mathbb{Q}_0^\Omega \,||\, \mathbb{P}_0^\theta) < +\infty$.*

The condition on f^θ is very mild, and it is satisfied if f^θ is bounded, as is the case for most neural networks.

3 LEARNING Ω-BRIDGE MODELS

Let $\{x^{(i)}\}_{i=1}^n$ be an i.i.d. sample from an unknown distribution Π^* on a domain $\Omega \subseteq \mathbb{R}^d$. Our goal is to learn the parameter θ for the Ω-bridge model $\mathbb{P}^\theta$ in (10) such that the terminal distribution $Z_T \sim \mathbb{P}_T^\theta$ matches the data $X \sim \Pi^*$. We should distinguish $\mathbb{P}^\theta$, which is the trainable generative model, and $\mathbb{Q}$, which is a fixed "baseline process" that helps us to derive methods for constructing and learning the model. $\mathbb{Q}$ can be the simple Brownian motion in Example 2.1 and 2.2.



As the case of other diffusion models, $\mathbb{P}^\theta$ can be viewed as a model with an infinite dimensional latent variable of the intermediate trajectories of $\boldsymbol{Z}$. Hence, a canonical learning approach is expectation maximization (EM), which alternates between

1) E-step: estimating the posterior $\mathbb{P}^{\theta,x} := \mathbb{P}^\theta(\boldsymbol{Z} \mid Z_T = x)$ of the latent trajectories $\boldsymbol{Z}$ given the observation $Z_T = x$;

2) M-step: estimating the parameter θ with $\boldsymbol{Z}$ imputed from $\mathbb{P}^{\theta,x}$.

A key challenge, however, is that the posterior distribution $\mathbb{P}^{\theta,x}$ is difficult to calculate due to the presence of neural force field in $\mathbb{P}^\theta$ (as the h-transform formula would have no closed form), and it need to be iteratively updated as θ changes. Following DDPM (Ho et al., 2020), we consider a simpler approach that replaces the posterior $\mathbb{P}^{\theta,x}$ with an arbitrary x-bridge, denoted by $\mathbb{Q}^x$. This yields a simplified EM algorithm without the expensive posterior inference in the E-step. A natural choice is the conditioned process $\mathbb{Q}^x := \mathbb{Q}(\boldsymbol{Z} \mid Z_T = x)$, but the method works for a general x-bridge.

Specifically, let $\mathbb{Q}^{\Pi^*}(\cdot) = \int \mathbb{Q}^x(\cdot)\Pi^*(\mathrm{d}x)$ be the mixture of the x-bridges whose end point x is randomly drawn from the data distribution $x \sim \Pi^*$. The trajectories from $\mathbb{Q}^{\Pi^*}$ can be generated in the following "backward" way: first drawing a data point $x \sim \Pi^*$, and then $\boldsymbol{Z} \sim \mathbb{Q}^x$ conditioned on the end point x. Obviously, by construction, the terminal distribution of $\mathbb{Q}^{\Pi^*}$ equals Π^*, that is, $\mathbb{Q}_T^{\Pi^*} = \Pi^*$. Then, the model $\mathbb{P}^\theta$ can be estimated by fitting data drawn from $\mathbb{Q}^{\Pi^*}$ using maximum likelihood estimation:

$$\min_\theta \left\{ \mathcal{L}(\theta) := \mathcal{KL}(\mathbb{Q}^{\Pi^*} \mid\mid \mathbb{P}^\theta) \right\}. \tag{11}$$

The classical (variational) EM would alternatively update θ (M-step) and $\mathbb{Q}^x$ (E-step) to make $\mathbb{Q}^x \approx \mathbb{P}^{\theta,x}$. Why is it OK to simply drop the E-step? At the high level, it is the benefit from using universal approximators like deep neural networks: if the model space of $\mathbb{P}^\theta$ is sufficiently rich, by minimizing the KL divergence in (11), $\mathbb{P}^\theta$ can approximate the given $\mathbb{Q}^{\Pi^*}$ well enough (in a way that is made precise in sequel) such that their terminal distributions are close: $\mathbb{P}_T^\theta \approx \mathbb{Q}_T^{\Pi^*} = \Pi^*$.

☞ *Learning latent variable models require no E-step if the model space is sufficiently rich.*

We should see that in this case the latent variables Z in the learned model $\mathbb{P}^\theta$ is *dictated* by the choice of the imputation distribution $\mathbb{Q}$ since we have $\mathbb{P}^{\theta,x} = \mathbb{Q}^x$ when the KL divergence in (11) is fully minimized to zero; EM also achieves $\mathbb{P}^{\theta,x} = \mathbb{Q}^x$ but has the imputation distribution $\mathbb{Q}^x$ determined by the model $\mathbb{P}^\theta$, not the other way.

Loss Function In its general form, the x-bridge $\mathbb{Q}^x$ that we use can be a non-Markov diffusion process

$$\mathbb{Q}^x: \quad \mathrm{d}Z_t = \eta^x(\boldsymbol{Z}, t)\mathrm{d}t + \sigma(Z_t, t)\mathrm{d}W_t, \quad Z_0 \sim \mu^x, \tag{12}$$

which has the same diffusion coefficient $\sigma(Z_t, t)$ as $\mathbb{P}^\theta$ in (10), and any x-dependent η^x and initialization μ^x once the x-bridge condition is ensured. As the general framework, we assume that η^x can depend on the whole trajectory $\boldsymbol{Z}$.

Using Girsanov theorem (e.g., Oksendal, 2013), with $\mathbb{P}^\theta$ in (10) and $\mathbb{Q}^x$ in (12), the KL divergence in (11) can be shown to equal to

$$\mathcal{L}(\theta) = \mathbb{E}_{\substack{x\sim\Pi^* \\ Z\sim\mathbb{Q}^x}} \Bigg[\underbrace{-\log p_0^\theta(Z_0)}_{\text{MLE of initial dist.}} + \frac{1}{2}\int_0^T \underbrace{\left\| \sigma^{-1}(Z_t, t)(s^\theta(Z_t, t) - \eta^x(\boldsymbol{Z}, t)) \right\|^2}_{\text{score matching}} \mathrm{d}t \Bigg] + const, \tag{13}$$

Algorithm 2 Learning Ω-Bridge Diffusion Models

Input: A dataset $\mathcal{D} := \{x^{(i)}\}$ drawn from distribution Π^* on a domain Ω.
Setup: Specify an x-bridge $\mathbb{Q}^x$ and an Ω-bridge $\mathbb{Q}^\Omega$

$$\mathbb{Q}^x : \ \mathrm{d}Z_t = \eta^x(\boldsymbol{Z}, t)\mathrm{d}t + \sigma(Z_t, t)\mathrm{d}W_t, \qquad \mathbb{Q}^\Omega : \ \mathrm{d}Z_t = \eta^\Omega(Z_t, t)\mathrm{d}t + \sigma(Z_t, t)\mathrm{d}W_t,$$

Specify the generative model $\mathbb{P}^\theta$ based on $\mathbb{Q}^\Omega$ and a neural network f^θ:

$$\mathbb{P}^\theta : \quad \mathrm{d}Z_t = (\sigma(Z_t, t)f^\theta(Z_t, t) + \eta^\Omega(Z_t, t))\mathrm{d}t + \sigma(Z_t, t)\mathrm{d}W_t, \quad Z_0 \sim \mathbb{P}_0^\theta.$$

Default: let $\mathbb{Q}$ be the law of $\mathrm{d}Z_t = \sigma_t \mathrm{d}W_t$ and derive the bridges by h-transform as $\mathbb{Q}^x = \mathbb{Q}(\cdot|Z_T = x)$ in Eq (7) and $\mathbb{Q}^\Omega = \mathbb{Q}(\cdot|Z_T \in \Omega)$ in Eq (9).
Training: Estimating θ by minimizing the loss function (13) using any off-the-shelf optimizer.
Sampling: Generate sample Z_T from $\mathbb{P}^\theta$ with the trained parameter θ.

where we write s^θ as the overall drift force of $\mathbb{P}^\theta$ in (10), that is,

$$s^\theta(z, t) = \sigma(z, t)f^\theta(z, t) + \eta^\Omega(z, t),$$

and p_0^θ is the probability density function (PDF) of the initial distribution $\mathbb{P}_0^\theta$. Therefore, $\mathcal{L}(\theta)$ is a sum of the negative log-likelihood of the initial distribution that encourages $\mathbb{P}_0^\theta \approx \mathbb{Q}_0^{\Pi^*}$, and a least squares loss between s^θ and η^x. In practice, we simply fix the initial distribution $\mathbb{P}_0^\theta$ to be a delta measure on a fixed point (say $x_0 = 0$), so we only need to train the drift function f^θ. Algorithm 1 shows a simple instance of the framework when the baseline process $\mathbb{Q}$ is the standard Brownian motion $\mathrm{d}Z_t = \mathrm{d}W_t$, $\mathbb{Q}^x = \mathbb{Q}(\cdot \mid Z_T = x)$ and $\sigma(z, t) = 1$. Note that the least squares term in (13) can be viewed as enforcing $f^\theta \approx \sigma^{-1}(\eta^x - \eta^\Omega)$, which reduces to $f^\theta \approx \nabla \log \omega^\Omega$ in the case of Algorithm 1.

Related Works Bridge processes provide a simple and flexible approach to learning diffusion generative models, which was explored in (Peluchetti, 2021; Ye et al., 2022; Wu et al., 2022), Our main focus is to extend and specialize the bridge-based framework to a constrained domain Ω, and hence significantly extend the scope of diffusion models.

In general, our approach can be applied to any domain Ω on which an Ω-bridge can be derived (using h-transform or any other method), and hence provides a highly flexible framework for learning with structured data (be it discrete, continuous, or their mixes). This distinguishes it with existing approaches that are designed for special types of data (e.g., Ho et al. (2020); Hoogeboom et al. (2021); Austin et al. (2021) for discrete data). One closely related work is Ye et al. (2022), which proposes to learn first hitting diffusion models for generation on both discrete sets and spheres. The advantage of our approach is that it is simpler and easier to derive for more complex types of domains.

4 Experiments

We evaluate our algorithms for generating mixed-typed tabular data, grid-valued point clouds, categorical semantic segmentation maps, discrete CIFAR10 images. We observe that Ω-bridge provides a particularly attractive and superb approach to generating data from various constrained domains.

Algorithm Overview For all experiments, we use Algorithm 2 with the default choice of $\mathbb{Q}^x$ in (7) and $\mathbb{Q}^\Omega$ in (9). The specific form of η^Ω is derived based on the specific choice of the domain Ω. By default, we set the initialization $Z_0 = 0$ and the optimizer Adam.

4.1 Generating Mixed-Type Tabular Data

Learning to generate tabular data is challenging, because tabular data usually contains a mixture of discrete and continuous attributes (Xu et al., 2019; Park et al., 2018). Unlike carefully designing special GANs as in previous works (Xu et al., 2019; Srivastava et al., 2017), Ω-bridge can be seamlessly applied to mixed-typed tabular data generation without any further modification. In contrast,

	Logistic (↑)	AdaBoost (↑)	MLP (↑)
Real Training Data	0.877±0.021	0.912±0.013	0.897±0.012
TVAE (Xu et al., 2019)	0.825±0.012	0.876±0.005	0.845±0.008
CTGAN (Xu et al., 2019)	0.649±0.014	0.841±0.021	0.843±0.016
CopulaGAN (Patki et al., 2016)	0.683±0.015	0.859±0.004	0.853±0.009
Mixed-Bridge	**0.868±0.010**	**0.884±0.005**	**0.877±0.006**

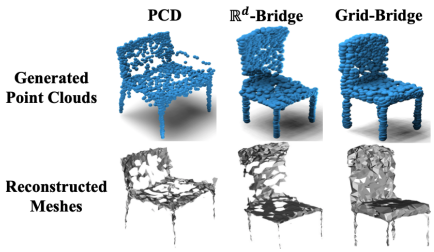
Table 1: Classification accuracy on the Adult Income dataset with different classifiers when trained with data synthesized by generative models. Real Training Data shows the upper bound of the metrics.

diffusion processes that solely work on discrete domain (Austin et al., 2021; Hoogeboom et al., 2021) cannot be applied to this task.

In this experiment, we use the *Adult Income* dataset (Kohavi, 1996), which contains 30,162 training samples. The data points are described by a series of attributes, including continuous (`age`, `capital-gain`, etc.) and discrete (`sex`, `race`, etc.). We compare with conditional tabular GAN (Xu et al., 2019) (CTGAN), CopulaGAN (Patki et al., 2016), and Table VAE (Xu et al., 2019) (TVAE), which are state-of-the-art GAN-based and VAE-based generative models for mixed-typed tabular data. Following previous works (Xu et al., 2019; Patki et al., 2016), we measure the classification accuracy on the real data of logistic regression, AdaBoost classifier and MLP classifier when trained on the generated data.

In the Ω-bridge model, we set $\sigma_t = 3\exp(-3t)$ and f^θ a 3-layer MLP. In this case, $\Omega = I_1 \times \cdots \times I_{15}$, where I_1 to I_9 are discrete domains and I_{10} to I_{15} are non-negative continuous domains. For discrete domains, $I = \{e_1, \ldots, e_d\}$, we have, $\eta^I(z,t) = \sigma_t^2 \nabla_z \log \sum_{e \in I} \exp\left(-\frac{\|z-e\|^2}{2(\beta_T - \beta_t)}\right)$; for non-negative continuous domains, $I = [0, +\infty)$, derivation shows $\eta^I(z,t) = \sigma_t^2 \nabla_z \log\left(F(\frac{z}{\sqrt{\beta_T - \beta_t}})\right)$, where F is the standard Gaussian CDF. Finally, we have $\eta^\Omega(z,t) = \sum_{i=1}^{15} \eta^{I_i}(z_i, t)$ for the whole domain Ω. We set the number of diffusion steps to $K = 2000$. Results are shown in Table 1.

Result All the three different classifiers yield the highest accuracy when trained on the data generated by our method, referred to as *Mixed-Bridge* in this case. The result reflects that the data generated by Mixed-Bridge is closer to the real distribution than the baseline methods.



Method	MMD ↓	COV ↑	1-NNA ↓
PCD (Luo & Hu, 2021a)	13.37	46.60	58.94
$\mathbb{R}^d$-Bridge	13.30	46.52	59.32
Grid-Bridge	**12.85**	**47.78**	**56.25**

Figure 2 & Table 2: The point clouds (upper row) generated by different methods and meshes reconstructed from them (lower row). Grid-Bridge obtains more uniform points and hence better mesh thanks to the integer constraints. Numbers in the table are multiplied by 10^3.

4.2 Generating Integer-valued Point Clouds

A feature of point clouds in 3D objects in graphics is that they tend to distribute evenly, especially if they are discretized from a mesh. This aspect is omitted in most existing works on point cloud generation. As a result they tend to generate non-uniform points that are unsuitable for real applications, which often involve converting back to meshes with procedures like Ball-Pivoting (Bernardini et al., 1999). We apply our method to generate point clouds that constrained on a integer grid which we show yields much more uniformly distributed points. To the best of our knowledge, we are the first work on integer-valued 3D point cloud generation.

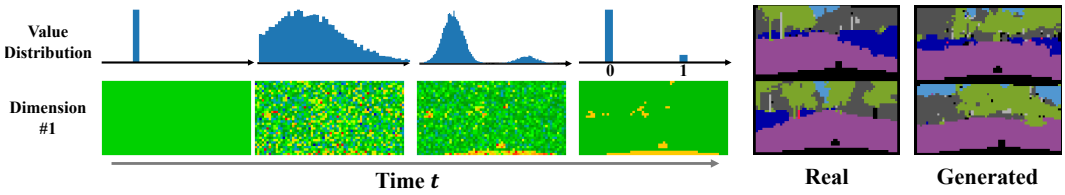


Figure 3: Results on generating categorical segmentation maps. Each pixel here an `one-hot` vector. Each dimension of the Ω-bridge starts from a deterministic and evolve through a stochastic trajectory to converge to either 0 or 1. The generated samples have similar visual quality to the training data.

Methods	ELBO (↓)	IWBO (↓)
Uniform Dequantization (Uria et al., 2013)	1.010	0.930
Variational Dequantization (Ho et al., 2019)	0.334	0.315
Argmax Flow (Softplus thres.) (Hoogeboom et al., 2021)	0.303	0.290
Argmax Flow (Gumbel distr.) (Hoogeboom et al., 2021)	0.365	0.341
Argmax Flow (Gumbel thres.) (Hoogeboom et al., 2021)	0.307	0.287
Multinomial Diffusion (Hoogeboom et al., 2021)	0.305	-
Cat.-Bridge (Constant Noise)	0.844	0.707
Cat.-Bridge (Noise Decay A)	**0.276**	**0.232**
Cat.-Bridge (Noise Decay B)	0.301	0.285
Cat.-Bridge (Noise Decay C)	0.363	0.302

Table 3: Results on the CityScapes dataset. Cat. refers to 'Categorical'.

A point cloud is a set of points $\{x_i\}_{i=1}^m$, $x_i \in \mathbb{R}^3$ in the 3D space, where m refers to the number of points. We apply two variants of our method: $\mathbb{R}^d$-Bridge and Grid-Bridge. $\mathbb{R}^d$-Bridge generates points in the continuous 3D space, i.e., $\Omega_{\mathbb{R}} = \mathbb{R}^{3m}$. Grid-Bridge generate points that on integer grids, $\Omega_{\text{Grid}} = \{1, \dots, 128\}^{3m}$. We fix the diffusion coefficient $\sigma_t = 1$. The number of diffusion steps K is set to 1000. We test our method on ShapeNet (Chang et al., 2015) chair models, and compare it with Point Cloud Diffusion (PCD) (Luo & Hu, 2021a), a state-of-the-art continuous diffusion-based generative model for point clouds. The neural network f^θ in our methods are the same as that of PCD for fair comparison. Qualitative results and quantitative results are shown in Figure 2 and Table 2. As common practice (Luo & Hu, 2021a,b), we measure minimum matching distance (MMD), coverage score (COV) and 1-NN accuracy (1-NNA) using Chamfer Distance (CD) with the test dataset.

Result Both $\mathbb{R}^d$-Bridge and Grid-Bridge get better MMD, COV, and 1-NNA than PCD. Moreover, by constraining the domain of interest to the integer grids, Grid-Bridge yields even better performance than $\mathbb{R}^d$-Bridge. In Figure 2, since the point clouds generated by Grid-Bridge are limited to integer grids, the reconstructed meshes from Ball-Pivoting clearly have higher quality than $\mathbb{R}^d$-Bridge and PCD.

4.3 Generating Semantic Segmentation Maps on CityScapes

We consider unconditionally generating categorical semantic segmentation maps. We represent each pixels as a `one-hot` categorical vector. Hence the data domain is $\Omega = \{e_1, \dots, e_c\}^{h \times w}$, where c is the number of classes and e_i is the i-th c-dimensional one-hot vector, and h, w represent the height and width of the image. In CityScapes (Cordts et al., 2016), $h = 32, w = 64, c = 8$. In this experiment, we test different schedule of the diffusion coefficient σ_t, including *(Constant Noise)*: $\sigma_t = 1$; *(Noise Decay A)*: $\sigma_t = a \exp(-bt)$; *(Noise Decay B)*: $\sigma_t = a(1-t)$; *(Noise Decay C)* $\sigma_t = a(1 - \exp(-b(1-t)))$. Here a and b are hyper-parameters. The number of diffusion steps K is set to 500. We measure the negative log-likelihood (NLL) of the test set using the learned models. The NLL (bits-per-dimension) is estimated with evidence lower bound (ELBO) and importance weighted bound (IWBO) (Burda et al., 2016), respectively, as in (Hoogeboom et al., 2021). We compare Ω-Bridge with a state-of-the-art categorical diffusion algorithm, Argmax Flow (and Multinomial Diffusion) (Hoogeboom et al., 2021), and the traditional methods, uniform dequantization (Uria et al., 2013) and variational dequantization (Ho et al., 2019). The numerical results of the baselines are directly adopted from (Hoogeboom et al., 2021), and experiment configuration is kept the same

Methods	IS (↑)	FID (↓)	NLL (↓)
Discrete			
D3PM uniform L_{vb} (Austin et al., 2021)	5.99	51.27	5.08
D3PM absorbing L_{vb} (Austin et al., 2021)	6.26	41.28	4.83
D3PM Gauss L_{vb} (Austin et al., 2021)	7.75	15.30	3.966
D3PM Gauss $L_{\lambda=0.001}$ (Austin et al., 2021)	8.54	8.34	3.975
D3PM Gauss + logistic $L_{\lambda=0.001}$	8.56	7.34	3.435
Integer-Bridge (Init. A)	**8.77**	**6.77**	3.46
Integer-Bridge (Init. B)	8.68	6.91	**3.35**
Integer-Bridge (Init. C)	8.72	6.94	3.40

Table 4: Discrete CIFAR10 Image Generation

for fair comparison. The neural network f^θ is the same as (Hoogeboom et al., 2021). The results are shown in Figure 3 and Table 3. Our Ω-bridge is named *Categorical-Bridge (Cat.-Bridge)* in this experiment.

Result We observe that all the four kinds of Cat.-Bridge can successfully generate categorical semantic segments, and different noise schedules result in different empirical performance. Among the four variants of Cat.-Bridge, Cat.-Bridge with Noise Decay A yields the best ELBO and IWBO, surpassing all the other algorithms in comparison.

4.4 Generating Discrete CIFAR10 Images

In this experiment, we apply three types of bridges. All of these bridges use the same output domain $\Omega = \{0, \ldots, 255\}^{h \times w \times c}$, where h, w, c are the height, width and number of channels of the images, respectively. We set $\sigma_t = 3\exp(-3t)$. We consider different initial distributions: *(Init. A)* $Z_0 = 128$; *(Init. B)* $Z_0 = \hat{\mu}_0$, *(Init. C)* $Z_0 \sim \mathcal{N}(\hat{\mu}_0, \hat{\sigma}_0)$, where $\hat{\mu}_0$ and $\hat{\sigma}_0$ are the empirical mean and variance of pixels in the CIFAR10 training set. The number of diffusion steps K is set to 1000. We compare with the variants of a state-of-the-art discrete diffusion model, D3PM (Austin et al., 2021). For fair comparison, we use the DDPM backbone (Ho et al., 2020) as the neural drift f^θ in our method, similar to D3PM. We report the Inception Score (IS) Salimans et al. (2016), Fréchet Inception Distance (FID) Heusel et al. (2017) and negative log-likelihood (NLL) of the test dataset. We call our method *Integer-Bridge* in this case. The results are shown in Table 4 and Figure 4.

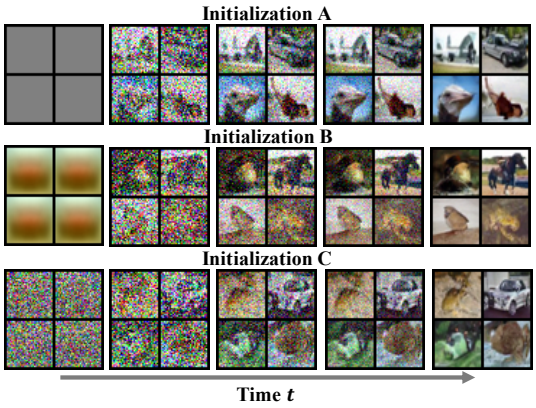


Figure 4: Integer-bridges can generate high-quality discrete samples with different initial distribution.

Result In Table 4, Integer-Bridge with Initialization A,B,C can all get lower FIDs (≤ 7) than the variants of D3PM. Among the three kinds of Integer-bridges, Integer-Bridge (Init. B) obtains the lowest NLL (3.35). It also beats D3PM Gauss + logistic (3.435) on NLL, which has the best NLL in the variants of D3PM.

5 Conclusion and Limitations

We present a framework for learning diffusion generative models that enables algorithmic extensions to structured data domains. It leaves a number of directions for further explorations and improvement. For example, the practical impact of the choices of the bridges $\mathbb{Q}$, in terms of initialization, dynamics, and noise schedule, are still not well understood and need more systematical studies. Moreover, application of Ω-bridge to many other practical fields, e.g., audio synthesis and language generation, also needs investigation in the future.

REFERENCES

Jacob Austin, Daniel D Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. Structured denoising diffusion models in discrete state-spaces. *Advances in Neural Information Processing Systems*, 34:17981–17993, 2021.

Fausto Bernardini, Joshua Mittleman, Holly Rushmeier, Cláudio Silva, and Gabriel Taubin. The ball-pivoting algorithm for surface reconstruction. *IEEE transactions on visualization and computer graphics*, 5(4):349–359, 1999.

Yuri Burda, Roger B Grosse, and Ruslan Salakhutdinov. Importance weighted autoencoders. In *ICLR (Poster)*, 2016.

Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015.

Nanxin Chen, Yu Zhang, Heiga Zen, Ron J Weiss, Mohammad Norouzi, and William Chan. Wavegrad: Estimating gradients for waveform generation. In *International Conference on Learning Representations*, 2020.

Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 34, 2021.

Joseph L Doob. *Classical potential theory and its probabilistic counterpart*, volume 549. Springer, 1984.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

Jonathan Ho, Xi Chen, Aravind Srinivas, Yan Duan, and Pieter Abbeel. Flow++: Improving flow-based generative models with variational dequantization and architecture design. In *International Conference on Machine Learning*, pp. 2722–2730. PMLR, 2019.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

Jonathan Ho, Chitwan Saharia, William Chan, David J Fleet, Mohammad Norouzi, and Tim Salimans. Cascaded diffusion models for high fidelity image generation. *Journal of Machine Learning Research*, 23(47):1–33, 2022.

Emiel Hoogeboom, Didrik Nielsen, Priyank Jaini, Patrick Forré, and Max Welling. Argmax flows and multinomial diffusion: Learning categorical distributions. *Advances in Neural Information Processing Systems*, 34, 2021.

Ron Kohavi. Scaling up the accuracy of naive-bayes classifiers: A decision-tree hybrid. In *Kdd*, volume 96, pp. 202–207, 1996.

Zhifeng Kong and Wei Ping. On fast sampling of diffusion probabilistic models. In *ICML Workshop on Invertible Neural Networks, Normalizing Flows, and Explicit Likelihood Models*, 2021.

Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. Diffwave: A versatile diffusion model for audio synthesis. In *International Conference on Learning Representations*, 2020.

Antoine Lejay. The girsanov theorem without (so much) stochastic analysis. In *Séminaire de Probabilités XLIX*, pp. 329–361. Springer, 2018.

Christian Léonard, Sylvie Rœlly, and Jean-Claude Zambrini. Reciprocal processes. a measure-theoretical point of view. *Probability Surveys*, 11:237–269, 2014.

Xingchao Liu, Xin Tong, and Qiang Liu. Sampling with trusthworthy constraints: A variational gradient framework. *Advances in Neural Information Processing Systems*, 34:23557–23568, 2021.

Shitong Luo and Wei Hu. Diffusion probabilistic models for 3d point cloud generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2837–2845, 2021a.

Shitong Luo and Wei Hu. Score-based point cloud denoising. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4583–4592, 2021b.

Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, pp. 8162–8171. PMLR, 2021.

Bernt Oksendal. *Stochastic differential equations: an introduction with applications*. Springer Science & Business Media, 2013.

Noseong Park, Mahmoud Mohammadi, Kshitij Gorde, Sushil Jajodia, Hongkyu Park, and Youngmin Kim. Data synthesis based on generative adversarial networks. *Proceedings of the VLDB Endowment*, 11(10), 2018.

N. Patki, R. Wedge, and K. Veeramachaneni. The synthetic data vault. In *2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, pp. 399–410, Oct 2016. doi: 10.1109/DSAA.2016.49.

Stefano Peluchetti. Non-denoising forward-time diffusions. 2021.

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022.

L Chris G Rogers and David Williams. *Diffusions, Markov processes and martingales: Volume 2, Itô calculus*, volume 2. Cambridge university press, 2000.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *Advances in neural information processing systems*, 29, 2016.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2020a.

Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in Neural Information Processing Systems*, 32, 2019.

Yang Song and Stefano Ermon. Improved techniques for training score-based generative models. *Advances in neural information processing systems*, 33:12438–12448, 2020.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2020b.

Yang Song, Conor Durkan, Iain Murray, and Stefano Ermon. Maximum likelihood training of score-based diffusion models. *Advances in Neural Information Processing Systems*, 34, 2021.

Akash Srivastava, Lazar Valkov, Chris Russell, Michael U Gutmann, and Charles Sutton. Veegan: Reducing mode collapse in gans using implicit variational learning. *Advances in neural information processing systems*, 30, 2017.

Benigno Uria, Iain Murray, and Hugo Larochelle. Rnade: The real-valued neural autoregressive density-estimator. *Advances in Neural Information Processing Systems*, 26, 2013.

Lemeng Wu, Chengyue Gong, Xingchao Liu, Mao Ye, and Qiang Liu. Diffusion-based molecule generation with informative prior bridges. *arXiv preprint arXiv:2209.00865*, 2022.

Zhisheng Xiao, Karsten Kreis, and Arash Vahdat. Tackling the generative learning trilemma with denoising diffusion gans. *arXiv preprint arXiv:2112.07804*, 2021.

Lei Xu, Maria Skoularidou, Alfredo Cuesta-Infante, and Kalyan Veeramachaneni. Modeling tabular data using conditional gan. *Advances in Neural Information Processing Systems*, 32, 2019.

Mao Ye, Lemeng Wu, and Qiang Liu. First hitting diffusion models. *arXiv preprint arXiv:2209.01170*, 2022.

Linqi Zhou, Yilun Du, and Jiajun Wu. 3d shape generation and completion through point-voxel diffusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5826–5835, 2021.

Roadmap The appendix is structured as follows:
• Appendix A provides the theoretical analysis and derivation of diffusion bridges. In particular, Appendix A.1 shows the derivation of the main training loss; Appendix A.2 explains why we can use arbitrary initialization when constructing bridge processes and discusses the reciprocal structure of $\mathbb{Q}^{\Pi^*}$; Appendix A.3 and A.4 presents details on the condition and examples of Ω-bridge construction.
• Appendix B shows additional experiment details and results.

A Theoretical Analysis on Bridges

A.1 Derivation of the main loss in Equation (13)

[Proof of Equation (13)] Denote by $\mathbb{Q}^x = \mathbb{Q}(\cdot|Z_\tau = x)$. Note that

$$\begin{aligned}\mathcal{KL}(\mathbb{Q}^{\Pi^*} \,||\, \mathbb{P}^\theta) &= \mathbb{E}_{x\sim\Pi^*, Z\sim\mathbb{Q}^x}\left[\log \frac{\mathrm{d}\mathbb{Q}^{\Pi^*}}{\mathrm{d}\mathbb{P}^\theta}(Z)\right] \\ &= \mathbb{E}_{x\sim\Pi^*, Z\sim\mathbb{Q}^x}\left[\log \frac{\mathrm{d}\mathbb{Q}^{x}}{\mathrm{d}\mathbb{P}^\theta}(Z) + \log \frac{\mathrm{d}\mathbb{Q}^{\Pi^*}}{\mathrm{d}\mathbb{Q}^x}(Z)\right] \\ &= \mathbb{E}_{x\sim\Pi^*}\left[\mathcal{KL}(\mathbb{Q}^x \,||\, \mathbb{P}^\theta)\right] + const,\end{aligned}$$

where $const$ denotes a constant that is independent of θ. Recall that $\mathbb{Q}^x$ follows $\mathrm{d}Z_t = \eta^x(Z_{[0,t]}, t)\mathrm{d}t + \sigma(Z_t, t)\mathrm{d}W_t$, and $\mathbb{P}^\theta$ follows $\mathrm{d}Z_t = s^\theta(Z_t, t)\mathrm{d}t + \sigma(Z_t, t)\mathrm{d}W_t$. By Girsanov theorem (e.g., Lejay, 2018),

$$\begin{aligned}\mathcal{KL}(\mathbb{Q}^x \,||\, \mathbb{P}^\theta) &= \mathcal{KL}(\mathbb{Q}_0^x \,||\, \mathbb{P}_0^\theta) + \frac{1}{2}\mathbb{E}_{Z\sim\mathbb{Q}^x}\left[\int_0^T \left\|s^\theta(Z_t, t) - \eta^x(Z_{[0,t]}, t)\right\|_2^2 \mathrm{d}t\right] \\ &= \mathbb{E}_{Z\sim\mathbb{Q}^x}\left[-\log p_0^\theta(Z_0) + \frac{1}{2}\int_0^T \left\|s^\theta(Z_t, t) - \eta^x(Z_{[0,t]}, t)\right\|_2^2 \mathrm{d}t\right] + const.\end{aligned}$$

Hence

$$\begin{aligned}L(\theta) &= \mathbb{E}_{x\sim\Pi^*, Z\sim\mathbb{Q}^x}\left[-\log p_0^\theta(Z_0) + \frac{1}{2}\int_0^T \left\|s^\theta(Z_t, t) - \eta^x(Z_{[0,t]}, t)\right\|_2^2 \mathrm{d}t\right] + const \\ &= \mathbb{E}_{Z\sim\mathbb{Q}^{\Pi^*}}\left[-\log p_0^\theta(Z_0) + \frac{1}{2}\int_0^T \left\|s^\theta(Z_t, t) - \eta^{Z_T}(Z_{[0,t]}, t)\right\|_2^2 \mathrm{d}t\right] + const.\end{aligned}$$

A.2 Markov and Reciprocal Properties of $\mathbb{Q}^{\Pi^*}$

Mixture of Bridges and Initialization It is an immediate observation that the mixtures of a set of bridges are also bridges: let $\mathbb{Q}^{z,A}$ be a set of A-bridges indexed by a variable z, then $\mathbb{Q}^A := \int \mathbb{Q}^{z,A}\mu(\mathrm{d}z)$ is an x-bridge for any distribution μ on z.

A special case is to take the mixture of the conditional bridges in (5) starting from different deterministic initialization, which shows that we can obtain a valid x-bridge by equipping the same drift in (5) with essentially *any* initialization. Hence, the choices of the drift force and initialization in $\mathbb{Q}^x$ can be completely decouple.

Proposition A.1. *Let $\tilde{\mathbb{Q}}$ is a path measure and Ω_x is the set of z for which $\tilde{\mathbb{Q}}^{z_0,x}(\cdot) := \tilde{\mathbb{Q}}(\cdot|Z_T = x, Z_0 = z_0)$ exists. Then $\mathbb{Q}^x := \int \tilde{\mathbb{Q}}^{z_0,x}\mu(\mathrm{d}z_0 \mid x)$ is an x-bridge, for any distribution μ on $\Omega \times \Omega$.*

[Proof of Proposition A.1] This is an obvious result. We have $\mathbb{Q}^{z_0,x}(Z_T = x) = 1$ by the definition of conditioned processes. Hence $\mathbb{Q}^x(Z_T = x) = \int \mathbb{Q}^{z_0,x}(Z_T = x)\mu(\mathrm{d}z_0 \mid x) = \int \mu(\mathrm{d}z_0 \mid x) = 1$.

Markov and Reciprocal Properties of $\mathbb{Q}^{\Pi^*}$ If $\mathbb{Q}^x$ is constructed as $\mathbb{Q}^x = \mathbb{Q}(\cdot|Z_T = x)$, it is easy to see that $\mathbb{Q}^{\Pi^*} := \int \mathbb{Q}^x(\cdot)\Pi^*(\mathrm{d}x)$ is Markov iff $\mathbb{Q}$ is Markov.

Proposition A.2. *Assume $\mathbb{Q}^x = \mathbb{Q}(\cdot \mid Z_T = x)$ and $\pi^*(z) := \frac{\mathrm{d}\Pi^*}{\mathrm{d}\mathbb{Q}_T}(z)$ exists and is positive everywhere. Then $\mathbb{Q}^{\Pi^*}$ is Markov, iff $\mathbb{Q}$ is Markov.*

[Proof of Proposition A.2] If $\mathbb{Q}^x = \mathbb{Q}(\cdot \mid Z_T = x)$, we have from the definition of $\mathbb{Q}^{\Pi^*}$:

$$\mathbb{Q}^{\Pi^*}(Z) = \mathbb{Q}(Z|Z_T)\Pi^*(Z_T) = \mathbb{Q}(Z)\pi^*(Z_T),$$

where $\pi^*(Z_T) = \frac{\mathrm{d}\Pi^*}{\mathrm{d}\mathbb{Q}_T}(Z_T)$. Therefore, $\mathbb{Q}^{\Pi^*}$ is obtained by multiplying a positive factor $\pi^*(Z_T)$ on the terminal state Z_T of $\mathbb{Q}$. Hence $\mathbb{Q}^{\Pi^*}$ has the same Markov structure as that of $\mathbb{Q}$.

If $\mathbb{Q}^x$ is constructed from mixtures of bridges as above, the resulting $\mathbb{Q}^{\Pi^*}$ is more complex. In fact, simply varying the initialization μ in Proposition (A.1) can change the Markov structure of $\mathbb{Q}^{\Pi^*}$.

Proposition A.3. *Take $\mathbb{Q}^x$ to be the dynamics in (7) initialized from $Z_0 \sim \mathcal{N}(0, v_0)$. Assume $\sigma_t > 0$, $\forall t \in [0, T]$. Then $\mathbb{Q}^{\Pi^*}$ is Markov only when $v_0 = 0$, or $v_0 = +\infty$.*

[Proof of Proposition A.3] When taking $\mathbb{Q}^x$ to be the dynamics (7) initialized from $Z_0 \sim \mu_0 = \mathcal{N}(0, v_0)$, we have $\mathbb{Q}^x = \int \mu_0(\mathrm{d}z_0)\tilde{\mathbb{Q}}^{z_0,x}$, where $\tilde{\mathbb{Q}}^{z_0,x} = \tilde{\mathbb{Q}}(\cdot|Z_0 = z_0, Z_T = x)$ with $\tilde{\mathbb{Q}}$ following Brownian motion $\mathrm{d}Z_t = \mathrm{d}W_t$. Hence, we can write $\mathbb{Q}^{\Pi^*}(\mathrm{d}Z) = \tilde{\mathbb{Q}}(\mathrm{d}Z)r(Z_0, Z_T)$, where $r(z_0, z_T) = \frac{\mathrm{d}\mu_0 \otimes \Pi^*}{\mathrm{d}\tilde{\mathbb{Q}}_{0,T}}(z_0, z_T)$. From Léonard et al. (2014), $\mathbb{Q}^{\Pi^*}$ is Markov iff $r(x, z_0) = f(x)g(z_0)$ for some f and g, which is not the case except the degenerated case ($v_0 = 0$ and $v_0 = +\infty$) because $\tilde{\mathbb{Q}}_{0,1}$ is not factorized.

On the other hand, when $v_0 = 0$, we have that $\mathbb{Q}^x = \mathbb{Q}(\cdot|Z_T = x)$ is the standard Brownian bridge and hence $\mathbb{Q}^{\Pi^*}$ is Markov following Proposition A.2. When $v_0 = +\infty$, as the case of SMLD, $\mathbb{Q}^{\Pi^*}$ is the law of $Z_t = \tilde{Z}_{T-t}$ with $\mathrm{d}\tilde{Z}_t = \mathrm{d}W_t$ and $\tilde{Z}_0 \sim \Pi^*$, which is also Markov.

The right characterization of $\mathbb{Q}^{\Pi^*}$ from Proposition (A.1) involves reciprocal processes (Léonard et al., 2014).

Definition A.4. *A process Z with law $\mathbb{Q}$ on $[0, T]$ is said to be reciporcal if it can be written into $\mathbb{Q} = \int \tilde{\mathbb{Q}}^{z_0,z_T}\mu(\mathrm{d}z_0, \mathrm{d}z_T)$, where $\tilde{\mathbb{Q}}$ is a Markov process and $\tilde{\mathbb{Q}}^{z_0,z_T} = \tilde{\mathbb{Q}}(\cdot|Z_0 = z_0, Z_T = z_T)$, and μ is a probability measure on $\Omega \times \Omega$.*

Proposition A.5. *$\mathbb{Q}^{\Pi^*}$ is reciprocal iff $\mathbb{Q}^x = \int \tilde{\mathbb{Q}}^{z_0,x}\mu(\mathrm{d}z_0 \mid x)$ for a Markov $\tilde{\mathbb{Q}}$ and distribution μ.*

[Proof of Proposition A.5] Note that

$$\mathbb{Q}^{\Pi^*}(\cdot) = \int \pi(\mathrm{d}x)\mathbb{Q}^x(\cdot) = \int \pi(\mathrm{d}x)\mu(\mathrm{d}z_0 \mid x)\tilde{\mathbb{Q}}^{z_0,x}(\cdot).$$

Hence if $\tilde{\mathbb{Q}}$ is Markov, $\mathbb{Q}^{\Pi^*}$ is reciprocal by Definition A.4.

On the other hand, if $\mathbb{Q}^{\Pi^*}$ is reciprocal, we have $\mathbb{Q}^{\Pi^*}(\cdot) = \int \mathbb{M}^{z_0,x}(\cdot)\mu(\mathrm{d}z_0, \mathrm{d}x)$ for some Markov process $\mathbb{M}$ and probability measure μ on $\Omega \times \Omega$. In this case, we have $\mathbb{Q}^x(\cdot) = \mathbb{Q}^{\Pi^*}(\cdot|Z_T = x) = \int \mathbb{M}^{z_0,x}(\cdot)\mu(\mathrm{d}z_0 \mid x)$, assuming it exits.

Intuitively, a reciprocal process can be viewed as connecting the head and tail of a Markov chain, yielding a single loop structure. A characteristic property is $\mathbb{Q}(X_{[s,t]} \in A \mid Z_{[0,s]}, Z_{[t,T]}) = \mathbb{Q}(X_{[s,t]} \in A \mid Z_s, Z_t)$, where A is any event that occur between time s and t. Solutions of the Schrodinger bridge problems are reciprocal processes (Léonard et al., 2014).

A.3 Condition for Ω-bridges

We provide the proof for Proposition 2.3.

[Proof of Proposition 2.3] By the formula of KL divergence between two diffusion processes, we have

$$\mathcal{KL}(\mathbb{Q}^\Omega \,||\mathbb{P}^\theta) = \mathcal{KL}(\mathbb{Q}_0^\Omega \,||\, \mathbb{P}_0^\theta) + \frac{1}{2}\mathbb{E}_{Z\sim\mathbb{Q}^\Omega}\left[\int_0^T \left\|f^\theta(Z_t,t)\right\|_2^2 \mathrm{d}t\right] < +\infty.$$

This means that $\mathbb{Q}^\Omega$ and $\mathbb{P}^\theta$ are absolutely continuous to each other, and hence have the same support. Therefore, $\mathbb{Q}^\Omega(Z_T \in \Omega) = 1$ implies that $\mathbb{P}^\theta(Z_T \in \Omega) = 1$.

A.4 Examples of Ω-Bridges

If Ω is a product space, the integration can be factorized into one-dimensional integrals. Specifically, assume $\Omega = I_1 \times \cdots I_d$, then

$$\eta^\Omega(z,t) = \left[\eta^{I_i}(z_i,t)\right]_{i=1}^d,$$

where η^{I_i} is the drift fore of the I_i-bridge, and z_i is the i-th element of $z = [z_i]$. Therefore, it is sufficient to focus on 1D case below.

Consider the bridge process constructed from the Brownian motion in (9). If Ω is a discrete set, say $\Omega = \{e_1 \ldots, e_K\}$, we have

$$\begin{aligned}\eta^\Omega(z,t) &= \sigma_t^2 \frac{1}{\sum_{k=1}^K \omega(e_k,z,t)} \sum_{k=1}^K \omega(e_k,z,t)\frac{e_k - z}{\beta_T - \beta_t} \\ &= \sigma_t^2 \nabla_z \log \sum_{k=1}^K \omega(e_k,z,t),\end{aligned}$$

where

$$\omega(e_k,z,t) = \exp\left(-\frac{\|z-e_k\|^2}{2(\beta_T-\beta_t)}\right).$$

If $\Omega = [a,b]$, we have

$$\begin{aligned}\eta^\Omega(z,t) &= \sigma_t^2 \frac{1}{\int_a^b \omega(e,z,t)} \int_a^b \omega(e,z,t)\frac{e-z}{\beta_T-\beta_t}\mathrm{d}e \\ &= \sigma_t^2 \nabla_z \log \int_a^b \omega(e,z,t)\mathrm{d}e \\ &= \sigma_t^2 \nabla_z \log\left(F(\frac{z-a}{\sqrt{\beta_T-\beta_t}}) - F(\frac{z-b}{\sqrt{\beta_T-\beta_t}})\right),\end{aligned}$$

where F is the standard Gaussian CDF.

B Additional Materials of the Experiments

In our experiments, $T = 1$ and $\epsilon = T/K = 1/K$. Moreover, we take the time grid by randomly sampling from $\{i/K\}_{i=0}^{K-1}$ for the training objective Eq. (13). For evaluation, we calculate the standard evidence lower bound (ELBO) by viewing the resulting time-discretized model as a latent variable model:

$$\mathbb{E}_{X\sim\Pi^*}[-\log \hat{p}_T^\theta(X)] \le \mathbb{E}_{Z\sim\mathbb{Q}^{\Pi^*}}\left[-\log\frac{\hat{p}_0^\theta(Z_0)}{q_0(Z_0)} - \sum_{k=1}^K \log\frac{\hat{p}^\theta_{t_{k+1}|t_k}(Z_{t_{k+1}}|Z_{t_k})}{q_{t_{k+1}|t_k}(Z_{t_{k+1}}|Z_{t_k})}\right],$$

where $t_k = (k-1)\epsilon$, and $\hat{p}^\theta$ is the density function of the time-discretized version of $\mathbb{P}^\theta$, and q is the density function of $\mathbb{Q}$. We adopt Monte-Carlo sampling to estimate the log-likelihood. As in (Song et al., 2020b), we repeat 5 times in the test set for the estimation. For categorical/integer/grid generation, the likelihood of the last step should take the rounding into account: in practice, we have $\hat{Z}_T = \mathrm{rounding}(\hat{Z}_{t_K} + \epsilon s^\theta(\hat{Z}_{t_K}, t_K) + \sqrt{\epsilon}\sigma(Z_{t_k}, t_K)\xi_K,\ \Omega)$, where $\mathrm{rounding}(x,\Omega)$ denotes finding the nearest element of x on Ω, and hence the likelihood $\hat{p}^\theta_{T|t_K}$ of the last step should incorporate the rounding operator as a part of the model.

age	workclass	fnlwgt	education	marital-status	occupation	relationship	race	sex	capital-gain	capital-loss	hours-per-week	native-country	income
55	Self-emp-not-inc	229791	Bachelors	Divorced	Prof-specialty	Other-relative	White	Male	2561	77	37	Cuba	>50K
45	Self-emp-not-inc	313479	Some-college	Never-married	Machine-op-inspct	Husband	White	Male	4660	483	38	United-States	>50K
81	Without-pay	106624	9th	Married-civ-spouse	Adm-clerical	Husband	Black	Male	1910	66	24	United-States	<=50K
41	Private	285037	9th	Divorced	Adm-clerical	Husband	White	Male	3754	46	28	Poland	<=50K
22	Private	21419	Bachelors	Widowed	Exec-managerial	Own-child	White	Female	4689	209	34	United-States	<=50K

Figure 5: Generated tabular data from Mixed-Bridge.

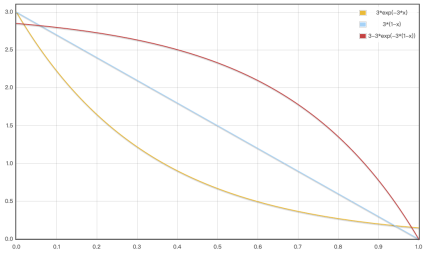


Figure 6: Visualization of the noise schedule of Noise decay A, Noise decay B and Noise decay C.

B.1 Generating Mixed-Type Tabular Data

In this experiment, the metrics are measured by the implementation from Synthetic Data Vault (Patki et al., 2016). For baseline methods, we adopt their open-sourced official implementation [1]. For the results in Table 1, we repeat the experiments with 5 different random seeds and report their standard deviation. We provide additional generated samples from Mixed-Bridge in Figure 5.

B.2 Generating Integer-valued Point Clouds

In this experiment, we need to process point cloud data on integer grid. To prepare the data, we firstly sample 2048 points from the ground truth mesh. Then, we normalize all the point clouds to a unit bounding box. After this, we simply project the points onto grid point by rounding the coordinate to integer. The metrics in the main text, MMD, COV and 1-NNA are computed with respect to the post-processed integer-valued training point clouds. For the results in Table 2, we repeat the experiments for 3 times and report the mean of the experiments.

B.3 Generating Semantic Segmentation Maps on CityScapes

In this experiment, we set *(Noise Decay A)*: $\sigma_t^2 = 3\exp(-3t)$; *(Noise Decay B)*: $\sigma_t^2 = 3(1-t)$; *(Noise Decay C)* $\sigma_t^2 = 3 - 3\exp(-3(1-t))$. We visualize the noise schedule in Figure 6. Note that, except for Constant Noise, all the other three processes gradually decrease the magnitude of the noise as $t \to 1$. For fair comparison, we use the same neural network as in Hoogeboom et al. (2021). The network is optimized with Adam optimizer with a learning rate of 0.0002. The model is trained for 500 epochs. The CityScapes dataset (Cordts et al., 2016) contains photos captured by the cameras on the driving cars. A pixel-wise semantic segmentation map is labeled for each photo. As in (Hoogeboom et al., 2021), we rescale the segmentation maps from cityscapes to 32×64 images using nearest neighbour interpolation. Our training set and test set is exactly the same as that of (Hoogeboom et al., 2021) for fair comparison. For the results in Table 3,we repeat the experiments for 3 times and report the mean of the experiments. We provide more samples in Figure 9.

B.4 Discrete CIFAR10 generation

The model is trained using the same training strategy as DDPM (Ho et al., 2020) with the code base provided in Song et al. (2020b). Specifically, the neural network is the same U-Net structure as

[1] https://github.com/sdv-dev/CTGAN

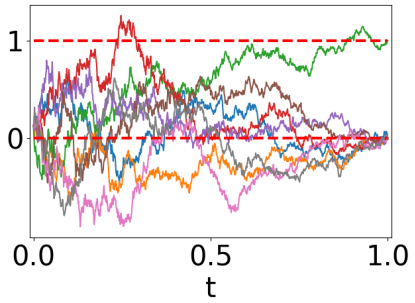
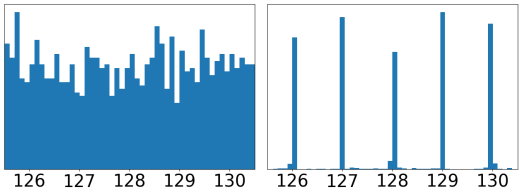


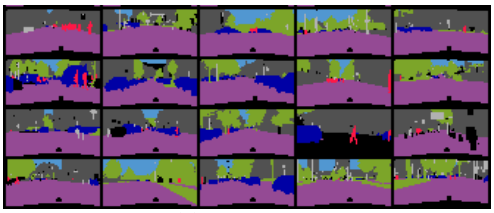
Figure 7: Diffusion process of one pixel (a 8-dimensional vector) in CityScapes. As $t \to 1$, 7 of the dimensions reaches 0, while 1 of the dimensions reaches 1, turning the vector into a `one-hot` vector.



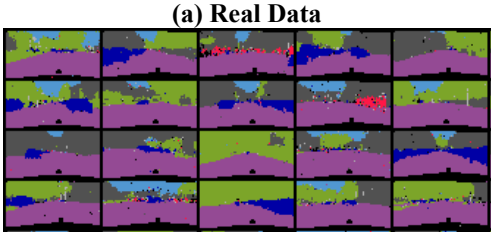
(a) Bridge-Continuous (b) Bridge-Integer

Figure 8: Final value distribution of the generated images with Bridge-Continuous and Bridge-Integer (before rounding) on CIFAR10. We only show the values in $[125.5, 130.5]$ for visual clarity. Integer-Bridge generates discrete values.

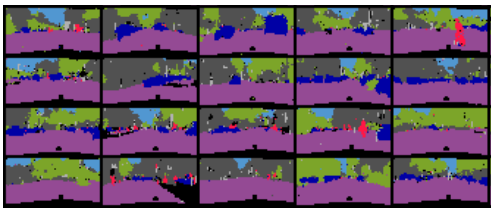
the implementation in Song et al. (2020b). The optimizer is Adam with a learning rate of 0.0002. According to common practice (Song & Ermon, 2020; Song et al., 2020b), the training is smoothed by exponential moving average (EMA) with a factor of 0.999. We use $K = 1000$ and $\mathrm{d}t = 0.001$ for discretizing the SDE. To account for the discretization error, after the final step, we apply rounding to the generated images to get real integer-valued images. We compare the value distribution of the generated images in Figure 8.



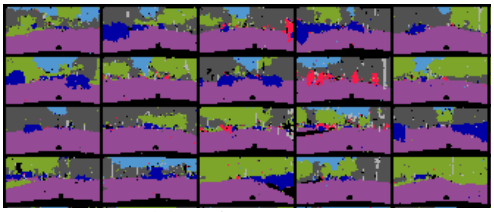
(a) Real Data



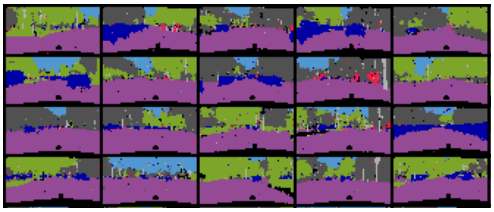
(b) Constant Noise



(c) Noise Decay A



(d) Noise Decay B



(e) Noise Decay C

Figure 9: Additional samples from real data, Constant Noise, Noise decay A, Noise decay B and Noise decay C.