Let us Build Bridges: Understanding and Extending Diffusion Generative Models

Anonymous Author(s)
Affiliation
Address
email

Abstract

Diffusion-based generative models have achieved promising results recently, but raise an array of open questions in terms of conceptual understanding, theoretical analysis, algorithm improvement and extensions to discrete, structured, non-Euclidean domains. This work tries to re-exam the overall framework, in order to gain better theoretical understandings and develop algorithmic extensions for data from arbitrary domains. By viewing diffusion models as latent variable models with unobserved diffusion trajectories and applying maximum likelihood estimation (MLE) with latent trajectories imputed from an auxiliary distribution, we show that both the model construction and the imputation of latent trajectories amount to constructing diffusion bridge processes that achieve deterministic values and constraints at end point, for which we provide a systematic study and a suit of tools. Leveraging our framework, we present a simple and unified approach to learning on data from different discrete and constrained domains. Experiments show that our methods perform superbly on generating images and semantic segments. The full paper is found at https://arxiv.org/abs/2208.14699.

1 Introduction

Diffusion-based deep generative models, notably score matching with Langevin dynamics (SMLD) [21, 22], denoising diffusion probabilistic models (DDPM) [8], and their variants [e.g., 23, 20, 11, 24, 16], have shown to achieve new state of the art results for image synthesis [5, 18, 9, 13], audio synthesis [3, 12], point cloud synthesis [14, 15, 27], and many other AI tasks. However, a range of open challenges arise on understanding, analyzing, and improving diffusion-based models. On the conceptual and theoretical perspective, existing methods have been derived from multiple angles, including denoising score matching [26, 21], time reversed diffusion [23], and variational bounds [8], but these approaches leave many design choices whose relations and effects have been unclear and difficult to analyze. On the practical side, standard approaches tend to be slow in both training and inference due to the need of a large number of diffusion steps, and are restricted to generating continuous data in $\mathbb{R}^d$ – special techniques such as dequantization [25, 7] and multinomial diffusion [10, 1] need to be developed case by case for different types of discrete data and the results still tend to be unsatisfying despite promising recent advances [10, 1].

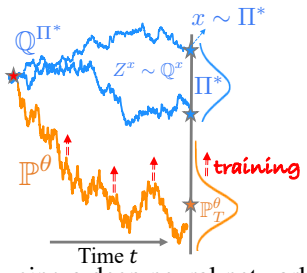
In this work, we approach diffusion models with a simple and classical statistical learning framework. By viewing the diffusion models as a latent variable model consisting of unobserved trajectories whose end points output observed data, the learning is decomposed into two parts: 1) constructing imputation mechanisms to generate latent trajectories that would have generated a given data point x, and 2) specifying and training the diffusion generative model to generate data on the domain Ω of interest by maximizing likelihood using the imputed trajectories. Both components involve constructing *diffusion bridge* processes, called x-bridge and Ω-bridge, whose end points guarantee to hit a deterministic value x or domain Ω at the terminal time, respectively. The design of learning

Under review at the NeurIPS 2022 Workshop on Score-Based Methods. Do not distribute.

algorithms reduces to constructing two bridges. Our framework allows us to decouple the various building blocks of the diffusion learning, algorithmic extensions to structured domains, and speedup in the regime of small sampling steps. Ω-bridge also provides a simple and universal approach to learning on data from an arbitrary domain Ω that can be embedded in $\mathbb{R}^d$ and on which the expectation of truncated standard Gaussian distribution can be evaluated. This includes product spaces of any type, bounded/unbounded, continuous/discrete, categorical/ordinal data, and their mix. The efficiency of the method is testified on a suit of examples, including generating images and segmentation maps.

2 Let us Build Bridges

2.1 Learning Latent Diffusion Models



Let $\{x^{(i)}\}_{i=1}^n$ be an i.i.d. sample from an unknown distribution Π^* on a domain $\Omega \subseteq \mathbb{R}^d$. We want to fit the data with a diffusion model $\mathbb{P}^\theta(\mathrm{d}Z)$, which specifies the distribution of a latent trajectory $Z = \{Z_t \colon t \in [0,T]\}$ that outputs an observation ($x = Z_T$) at the terminal time T. The evolution of Z is governed by an Ito process:

$$\mathrm{d}Z_t = s_t^\theta(Z_t)\mathrm{d}t + \sigma_t(Z_t)\mathrm{d}W_t, \quad \forall t \in [0,T], \qquad Z_0 \sim \mathbb{P}_0^\theta, \quad (1)$$

where W_t is a Wiener process; $\sigma \colon [0,T] \times \mathbb{R}^d \to \mathbb{R}^{d\times d}$ is a fixed, positive definite diffusion coefficient; the drift term $s^\theta \in [0,T] \times \mathbb{R}^d \to \mathbb{R}^d$ depends on a trainable parameter θ and is often specified using a deep neural network. The initial distribution $\mathbb{P}_0^\theta$ is often a fixed elementary distribution (Gaussian or deterministic). Here, $\mathbb{P}^\theta$ is the path measure on continuous trajectories Z following (1). We denote by $\mathbb{P}_t^\theta$ the marginal distribution of Z_t at time t. We want to estimate θ such that the terminal distribution $Z_T \sim \mathbb{P}_T^\theta$ matches the data $X \sim \Pi^*$.

If Ω is a strict subset of $\mathbb{R}^d$, e.g., bounded or discrete, then we need to specify the model $\mathbb{P}^\theta$ in (1) such that Z_T is guaranteed to arrive at Ω when $t = T$ (while the non-terminal states may not belong to Ω).

☞ *A process Z in $\mathbb{R}^d$ with law $\mathbb{P}$ is called a bridge to a set $B \subset \mathbb{R}^d$, or B-bridge, if $\mathbb{P}(Z_T \in B) = 1$.*

Following DDPM [8], we consider estimating θ with Z drawn from $\mathbb{Q}^x := \mathbb{Q}(Z|Z_T = x)$ of a pre-specified simple baseline process $\mathbb{Q}$. Here for each $x \in \Omega$, the conditioned process $\mathbb{Q}^x := \mathbb{Q}(Z|Z_T = x)$ is the distribution of the trajectories from $\mathbb{Q}$ that are pinned at x at time t. Therefore, $\mathbb{Q}^x$ is an x-bridge by definition. Let $\mathbb{Q}^{\Pi^*}(\cdot) = \int \mathbb{Q}^x(\cdot)\Pi^*(\mathrm{d}x)$ be the distribution of trajectories Z generated in the following *backward* way: first drawing a data point $x \sim \Pi^*$, and then $Z \sim \mathbb{Q}^x$ conditioned on the end point x. This construction ensures that 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 estimator: $\min_\theta \left\{\mathcal{L}(\theta) := \mathcal{KL}(\mathbb{Q}^{\Pi^*} \,||\, \mathbb{P}^\theta)\right\}$.

Loss Function Let us assume that $\mathbb{Q}^x$ yields a general non-Markov diffusion process of form

$$\mathrm{d}Z_t = \eta^x(Z_{[0,t]}, t)\mathrm{d}t + \sigma(Z_t, t)\mathrm{d}W_t, \qquad Z_0 \sim \mu^x, \quad (2)$$

where the drift η^x and initial distribution μ^x depend on the end point x and the diffusion coefficient σ is the same as that of $\mathbb{P}^\theta$. Here η^x can depend on the whole trajectory up to time t and hence $\mathbb{Q}^x$ can be non-Markov. $\mathbb{Q}^x$ is Markov if $\eta^x(Z_{[0,t]}, t) = \eta^x(Z_t, t)$. Using Girsanov theorem [e.g., 17], with $\mathbb{P}^\theta$ in (1) and $\mathbb{Q}^x$ in (2), the KL divergence can be reframed into a form of the score matching loss from [23, 24]:

$$\mathcal{L}(\theta) = \mathbb{E}_{\substack{x\sim\Pi^* \\ Z\sim\mathbb{Q}^x}}\left[\underbrace{-\log p_0^\theta(Z_0)}_{\text{MLE of initial dist.}} + \frac{1}{2}\int_0^T \underbrace{\left\|\sigma^{-1}(Z_t,t)(s^\theta(Z_t,t) - \eta^x(Z_{[0,t]},t))\right\|^2}_{\text{score matching}}\mathrm{d}t\right] + const, \quad (3)$$

As $\mathbb{P}^\theta$ is Markov by the model assumption, it can not perfectly fit $\mathbb{Q}^{\Pi^*}$ which is non-Markov in general. We resolve this by observing that it is not necessary to match the whole path measure ($\mathbb{P}^\theta \approx \mathbb{Q}^{\Pi^*}$) to match the terminal ($\mathbb{P}_T^\theta \approx \mathbb{Q}_T^{\Pi^*} = \Pi^*$). It is enough for $\mathbb{P}^\theta$ to be the best Markov approximation (a.k.a. Markovization) of $\mathbb{Q}^{\Pi^*}$, which matches all (hence terminal) fixed-time marginals with $\mathbb{Q}^{\Pi^*}$. Next we show how to derive $\eta^x(Z_{[0,t]}, t)$ for $\mathbb{Q}^{\Pi^*}$ and thus construct diffusion bridges of interest.

2.2 Bridge Construction

We discuss how to build bridges, both $\mathbb{Q}^x$ as x-bridges and $\mathbb{P}^\theta$ as Ω-bridges for constrained domains.

Constructing x-Bridges by h-transform Assume $\mathbb{Q}$ follows $\mathrm{d}Z_t = b(Z_t, t)\mathrm{d}t + \sigma(Z_t, t)\mathrm{d}W_t$. Then by using Doob's method of h-transforms [17], the conditioned process $\mathbb{Q}^x(\cdot) := \mathbb{Q}(\cdot \mid Z_T = x)$, if it exists, can be shown to be the law of

$$\mathrm{d}Z_t^x = \left(b(Z_t^x, t) + \sigma^2(Z_t^x, t)\nabla_z \log q_{T|t}(x \mid Z_t^x)\right)\mathrm{d}t + \sigma(Z_t^x, t)\mathrm{d}W_t, \quad Z_0 \sim \mathbb{Q}_{0|T}(\cdot \mid x), \tag{4}$$

where $q_{T|t}(x|z)$ is the density function of the transition probability $\mathbb{Q}_{T|t}(\mathrm{d}x|z) = \mathbb{Q}(Z_T \in \mathrm{d}x|Z_t = z)$, assuming it exists. The additional drift term $\sigma^2\nabla \log q_{T|t}(x|z)$ plays the role of steering Z_t towards the target $Z_T = x$. The initial distribution can be calculated by Bayes rule: $\mathbb{Q}_{0|T}(\mathrm{d}z|x) \propto \mathbb{Q}_0(\mathrm{d}z)q_{T|0}(x|z)$. We should note that the drift term in (4) is independent of the initialization $\mathbb{Q}_0$, which allows us to decouple in the choices of initialization and drift in bridges.

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

$$\mathrm{d}Z_t^x = \eta^x_{\mathrm{bb},\varsigma}(Z_t^x, t)\mathrm{d}t + \varsigma_t \mathrm{d}W_t, \qquad \textit{with} \quad Z_0^x \sim \mathbb{Q}_0^x = \mathbb{Q}_{0|T}(\cdot|x), \tag{5}$$

where $\mathbb{Q}_0^x(\mathrm{d}z) \propto \mathbb{Q}_0(\mathrm{d}z)\phi(x \mid z, \beta_T - \beta_t)$, and $\phi(\cdot|\mu, \sigma^2)$ is the density function of $\mathcal{N}(\mu, \sigma^2)$. $\eta^x_{\mathrm{bb},\varsigma}(Z_t^x, t) = \varsigma_t^2 \frac{x - Z_t^x}{\beta_T - \beta_t}$ is a Brownian bridge (BB) process. A simple case is when $\varsigma_t = 1$ and $\eta^x_{\mathrm{bb},1}(z, t) = \frac{x-z}{T-t}$.

$\mathbb{Q}_0$ is the initialization that can be arbitrarily set by the user. Two extreme choices of $\mathbb{Q}_0$ stand out: 1) The SMLD initialization can be viewed as the case when we initialize $\mathbb{Q}$ with an improper "uniform" prior $\mathbb{Q}_0 = 1$, corresponding $\mathbb{Q}_0^x = \mathcal{N}(0, v)$ with $v \to +\infty$. 2) Let z_0 be any point that can reach $Z_T = x$ under $\mathbb{Q}$ in that $x \in \mathrm{supp}(\mathbb{Q}_{T|0}(\cdot|z))$. If we take $\mathbb{Q}_0 = \delta_{z_0}$, the delta measure centered at z_0, the bridge $\mathbb{Q}^x$ has the same deterministic initialization $\mathbb{Q}_0^x = \delta_{z_0}$. Hence any deterministic initialization equipped with the drift in (4) yields a conditional bridge.

Constructing Ω-Bridges for Constrained Domains If Ω is a constrained domain, we need to specify the model $\mathbb{P}^\theta$ such that it is an Ω-bridge for any θ. We provide a simple method that works for *any domain on which integration of standard Gaussian density function can be calculated.* An importance class is product spaces of form $\Omega = I_1 \times I_2 \times \cdots I_d$, where I_i can be discrete sets or intervals in $\mathbb{R}$. Our method consists of two steps: 1) we first get a baseline Ω-bridge by deriving the conditioned process $\mathbb{Q}(\cdot \mid Z_T \in \Omega)$ from $\mathbb{Q}$ which by definition is an Ω-bridge; 2) we then show that add extra drifts on top of it keeps the Ω-bridge property unchanged.

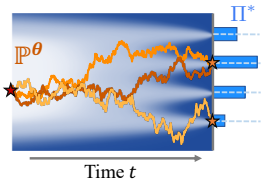


Figure 1: Ω-Bridges for discrete $\Omega = \{1, 2, 3, 4\}$.

In the first step, for any $\mathbb{Q}$ following $\mathrm{d}Z_t = b(Z_t, t)\mathrm{d}t + \sigma(Z_t, t)\mathrm{d}W_t$, the h-transform method shows that the conditioned process $\mathbb{Q}^\Omega := \mathbb{Q}(\cdot \mid Z_T \in \Omega)$ follows $\mathrm{d}Z_t = \eta^\Omega(Z_t, t)\mathrm{d}t + \sigma(Z_t, t)\mathrm{d}W_t$ with

$$\eta^\Omega(z, t) = b(z, t) + \sigma^2(z, t)\mathbb{E}_{x \sim \mathbb{Q}_{T|t,z,\Omega}}[\nabla_z \log q_{T|t}(x \mid z)], \qquad Z_0 \sim \mathbb{Q}_{0|T}(\cdot \mid X_T \in \Omega). \tag{6}$$

Its drift term is similar to that of the x-bridge in (4), except that x is now randomly drawn from an Ω-truncated transition probability: $\mathbb{Q}_{T|t,z,\Omega}(\mathrm{d}x \mid z) := \mathbb{Q}(Z_T = \mathrm{d}x \mid Z_t = z, Z_T \in \Omega)$. As an example, assuming $\mathbb{Q}$ follows $\mathrm{d}Z_t = \varsigma_t \mathrm{d}W_t$, we can show that $\mathbb{Q}^\Omega$ yields the following Ω-bridge:

$$\mathrm{d}Z_t = \eta^\Omega_{\mathrm{bb},\varsigma}(Z_t, t)\mathrm{d}t + \varsigma_t \mathrm{d}W_t, \qquad \eta^\Omega_{\mathrm{bb},\varsigma}(z, t) = \varsigma_t^2 \mathbb{E}_{x \sim \mathcal{N}_\Omega(z, \beta_T - \beta_t)}\left[\frac{x - Z_t}{\beta_T - \beta_t}\right], \tag{7}$$

where $\mathcal{N}_\Omega(z, \beta_T - \beta_t) = \mathrm{Law}(Z \mid Z \in \Omega)$ when $Z \sim \mathcal{N}(\mu, \sigma)$, which is an Ω-truncated Gaussian distribution. A general case is when $\Omega = I_1 \times \cdots I_d$, for which the expectation reduces to one dimensional Gaussian integrals; see Appendix for details.

In the second step, given an Ω-bridge $\mathbb{Q}^\Omega$, we construct a parametric model $\mathbb{P}^\theta$ by adding a learnable neural network f^θ in the drift:

$$\mathbb{P}^\theta: \qquad \mathrm{d}Z_t = (\sigma(Z_t, t)f^\theta(Z_t, t) + \eta^\Omega(Z_t, t))\mathrm{d}t + \sigma(Z_t, t)\mathrm{d}W_t, \quad Z_0 \sim \mathbb{P}_0^\theta. \tag{8}$$

By parameterizing $s^\theta(Z_t, t) = \sigma(Z_t, t)f^\theta(Z_t, t) + \eta^\Omega(Z_t, t)$, we obtain a Ω-bridge that guarantees to reach Ω with the help of the driving force $\eta^\Omega(Z_t, t)$.

3 Experiments

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 $\Omega = \{e_1, \ldots, e_c\}^{h \times w}$, 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 [4], $h = 32, w = 64, c = 8$. We test a number of bridge models with $\mathbb{Q} : \mathrm{d}Z_t = \varsigma_t \mathrm{d}W_t$ starting at the uniform point $Z_0 = 1/c$, with different schedule of the diffusion coefficient ς_t, including *(Constant Noise)*: $\varsigma_t = 1$; *(Noise Decay A)*: $\varsigma_t = a \exp(-bt)$; *(Noise Decay B)*: $\varsigma_t = a(1-t)$; *(Noise Decay C)* $\varsigma_t = a - a\exp(-b(1-t))$. Here a and b are hyper-parameters. 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) [2], respectively. The results are shown in Figure 2 and Table 1.

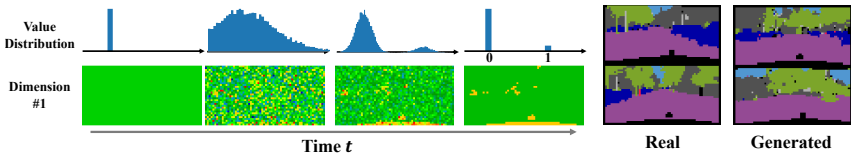


Figure 2: Results on generating categorical segmentation maps. Each pixel here an `one-hot` vector. Each dimension of the Ω-bridge starts from a deterministic and evolve through a stochastic trajectory to converge to either 0 or 1. The generated samples have similar visual quality to the training data.

Methods	ELBO (↓)	IWBO (↓)
Uniform Dequantization [25]	1.010	0.930
Variational Dequantization [7]	0.334	0.315
Argmax Flow (Softplus thres.) [10]	0.303	0.290
Argmax Flow (Gumbel distr.) [10]	0.365	0.341
Argmax Flow (Gumbel thres.) [10]	0.307	0.287
Multinomial Diffusion [10]	0.305	-
Bridge-Cat. (Constant Noise)	0.844	0.707
Bridge-Cat. (Noise Decay A)	**0.276**	**0.232**
Bridge-Cat. (Noise Decay B)	0.301	0.285
Bridge-Cat. (Noise Decay C)	0.363	0.302

Table 1: Results on the CityScapes dataset.

Methods	IS (↑)	FID (↓)	NLL (↓)
Discrete			
D3PM uniform L_{vb} [1]	5.99	51.27	5.08
D3PM absorbing L_{vb} [1]	6.26	41.28	4.83
D3PM Gauss L_{vb} [1]	7.75	15.30	3.966
D3PM Gauss $L_{\lambda=0.001}$ [1]	8.54	8.34	3.975
D3PM Gauss + logistic $L_{\lambda=0.001}$	8.56	7.34	3.435
Bridge-Integer (Init. A)	**8.77**	**6.77**	3.46
Bridge-Integer (Init. B)	8.68	6.91	**3.35**
Bridge-Integer (Init. C)	8.72	6.94	3.40

Table 2: Discrete CIFAR10 Image Generation

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 $\mathbb{Q}$ to be Brownian motion with the Noise Decay A in Section 3, that is, $\mathbb{Q} : \mathrm{d}Z_t = \varsigma_t \mathrm{d}W_t$, where $\varsigma_t = a\exp(-bt)$. We consider different initializations of $\mathbb{Q}$: *(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. We compare with the variants of a state-of-the-art discrete diffusion model, D3PM [1]. For fair comparison, we use the DDPM backbone [8] as the neural drift f^θ in our method, similar to D3PM. We report the Inception Score (IS) [19], Fréchet Inception Distance (FID) [6] and negative log-likelihood (NLL) of the test dataset. The results are shown in Table 2.

Generating Continuous CIFAR10 Images with Few-Step Diffusion Models In this experiment, we consider training diffusion models with very few sampling steps to generate continuous CIFAR10 images. For bridge, we use $\mathbb{Q} : Z_t = \mathrm{d}W_t$ initialized from $Z_0 = 0.5$. For SMLD, we use the implementation of NCSN++ in [23]. For DDPM, we use their original configuration. We use the DDPM backbone. We train the models with $K = 10, 20, 30, 40, 50$ diffusion steps. The results are shown in Table 3.

Methods	$K = 1000$	$K = 50$	$K = 40$	$K = 30$	$K = 20$	$K = 10$
DDPM	3.37	37.96	95.79	135.23	199.22	257.78
SMLD	**2.45**	140.98	157.67	169.62	267.21	361.23
Bridge	9.80	18.55	19.11	21.14	24.93	34.97
Bridge (Init. C)	9.65	**17.91**	**18.71**	**20.31**	**24.12**	**33.38**

Table 3: Results on continuous CIFAR10 generation when varying the number of diffusion steps in both training and testing. Our method shows significant advantages in regime of small diffusion steps ($K \leq 50$).

References

[1] 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.

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

[3] 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.

[4] 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.

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

[6] 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.

[7] 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*, pages 2722–2730. PMLR, 2019.

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

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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*, pages 2837–2845, 2021.

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

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

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

[18] 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.

[19] 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.

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

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

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

[23] 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*, 2020.

[24] 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.

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

[26] Pascal Vincent. A connection between score matching and denoising autoencoders. *Neural computation*, 23(7):1661–1674, 2011.

[27] 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*, pages 5826–5835, 2021.