On Density Estimation with Diffusion Models

Anonymous Author(s)
Affiliation
Address
email

Abstract

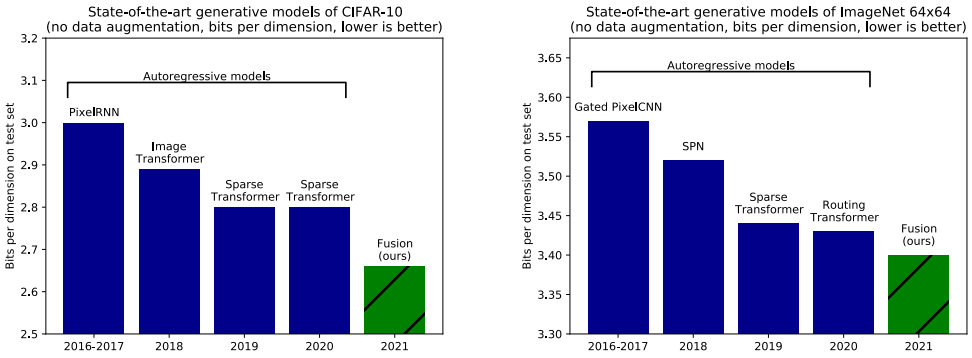
We introduce a flexible family of diffusion-based generative models that achieves state-of-the-art likelihoods on image density estimation benchmarks. Unlike other diffusion-based models, our method allows for efficient optimization of the noise schedule jointly with the rest of the model. We show that the evidence lower bound (ELBO) for our model simplifies to a remarkably short expression in terms of the signal-to-noise ratio of the diffusion process, thereby improving our theoretical understanding of this model class. Using this insight, we prove an equivalence between several models proposed in the literature. In addition, we show that the continuous-time ELBO is invariant to the noise schedule, except for the signal-to-noise ratio at its endpoints. This enables us to learn a noise schedule that minimizes the variance of the resulting ELBO estimator, leading to faster optimization. Combining these advances with architectural improvements, we obtain state-of-the-art likelihoods on the CIFAR-10 and ImageNet density estimation benchmarks, outperforming autoregressive models that have dominated these benchmarks for many years.

1 Introduction

Likelihood-based generative modeling is a central task in machine learning that is the basis for a wide range of applications ranging from speech synthesis [Oord et al., 2016], to translation [Sutskever et al., 2014], to compression [MacKay, 2003], to many others. Autoregressive models have long been the dominant model class on this task due to their tractable likelihood and expressivity, as shown in Figure 1. Diffusion models have recently shown impressive results in image [Ho et al., 2020, Song et al., 2021, Nichol and Dhariwal, 2021] and audio generation [Kong et al., 2020, Chen et al., 2020] in terms of perceptual quality, but have yet to match autoregressive models on density estimation benchmarks. In this paper we make several technical contributions that allow diffusion models to challenge the dominance of autoregressive models in this domain. Our main contributions are as follows:

- We introduce a flexible family of diffusion-based generative models that achieve new state-of-the-art log-likelihoods on standard image density estimation benchmarks (CIFAR-10 and ImageNet) without data augmentation. This is enabled by incorporating Fourier features into the diffusion model and using a learnable specification of the diffusion process, among other modeling contributions.
- We improve our theoretical understanding of density modeling using diffusion models by analyzing their evidence lower bound (ELBO), deriving a remarkably simple expression in terms of the signal-to-noise ratio of the diffusion process. This result delivers new insight into the model class: for the continuous time setting we prove a novel invariance of the generative model and its ELBO to the specification of the diffusion process, and we show that various diffusion models from the literature are equivalent up to a trivial rescaling of the data.

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.



(a) CIFAR-10

(b) ImageNet 64x64

Figure 1: Overview of state-of-the-art likelihood-based generative modeling methods in each of the past five years. In this paper, we propose a diffusion-based generative model, and efficient methods to train it. See Table 1 for more details.

2 Related Work

Our work builds on diffusion probabilistic models (DPMs) [Sohl-Dickstein et al., 2015], or *diffusion models* in short. DPMs can be viewed as a type of variational autoencoder (VAE) [Kingma and Welling, 2013, Rezende et al., 2014], whose structure and loss function allows for efficient training of arbitrarily deep models. Interest in diffusion models has recently reignited due to their impressive image generation results [Ho et al., 2020, Song and Ermon, 2020].

Ho et al. [2020] introduced a number of model innovations to the original DPM, with impressive results on image generation quality benchmarks. They showed that the ELBO objective, for a diffusion model with discrete time and diffusion variances shared across input dimensions, is equivalent to multi-scale denoising score matching, up to particular weightings per noise scale. Further improvements were proposed by Nichol and Dhariwal [2021], resulting in better log-likelihood scores.

Song and Ermon [2019] first proposed learning generative models through a multi-scale denoising score matching objective, with improved methods in Song and Ermon [2020]. This was later extended to continuous-time diffusion with novel sampling algorithms based on reversing the diffusion process [Song et al., 2021].

Earlier approaches fixed the diffusion process, and parameterized it in terms of the diffusion coefficients. For the fully general case, this would require integration to compute the mean and variance of $p(\mathbf{z}_t|\mathbf{z}_0)$, limiting its application to learned diffusion processes. In contrast, our approach directly parameterizes these means and variances, admitting efficient and exact computation of diffusion coefficients via automatic differentiation, while still allowing flexible learned diffusion processes.

Durkan and Song [2021] derive a maximum likelihood objective for continuous-time diffusion models, but this result holds for an intractable hypothetical model that arises from exact sampling from the actual model, followed by the forward diffusion process. In contrast, we derive the exact evidence lower bound (ELBO); see Appendix H for a comparison. Crucially, we show that what Durkan and Song [2021] claim is the maximum likelihood objective is actually the ELBO objective.

3 Model

We will focus on the most basic case of generative modeling, where we have a dataset of observations of $\mathbf{x}$, and the task is to estimate the marginal distribution $p(\mathbf{x})$. As with most generative models, the described methods can be extended to the case of multiple observed variables, and/or the task of estimating conditional densities $p(\mathbf{x}|\mathbf{y})$. The proposed latent-variable model consists of a diffusion process (Section 3.1) that we invert to obtain a hierarchical generative model (Section 3.2). We optimize the model parameters by maximizing the variational lower bound of the marginal log-likelihood (Section 4).

3.1 Forward time diffusion process

The starting point for our generative model is a diffusion process that begins with the data $\mathbf{x}$, and then samples a sequence of latent variables $\mathbf{z}_t$ given $\mathbf{x}$, where t runs forward in time from $t = 0$ to $t = 1$. The distribution of latent variable $\mathbf{z}_t$ conditioned on $\mathbf{x}$, for any $t \in [0, 1]$ is given by:

$$q(\mathbf{z}_t|\mathbf{x}) = \mathcal{N}\left(\alpha_t \mathbf{x}, \sigma_t^2 \mathbf{I}\right), \tag{1}$$

where α_t and σ_t^2 are scalar-valued functions that define the mean and variance of the marginal distributions with domain $t \in [0, 1]$ and range $\mathbb{R}^+$. Both α_t and σ_t^2 are smooth, such that their derivatives with respect to time t are finite. Furthermore, their ratio α_t^2/σ_t^2 is strictly monotonically decreasing in t, such that $\alpha_t^2/\sigma_t^2 < \alpha_s^2/\sigma_s^2$ for any $t > s$.

The joint distribution of latent variables $(\mathbf{z}_s, \mathbf{z}_t, \mathbf{z}_u)$, at subsequent timesteps $0 \leq s < t < u \leq 1$ are distributed as a first-order Markov chain, such that $q(\mathbf{z}_u|\mathbf{z}_t, \mathbf{z}_s) = q(\mathbf{z}_u|\mathbf{z}_t)$. The distribution of $\mathbf{z}_t$ given $\mathbf{z}_s$, for any $0 \leq s < t \leq 1$, is then:

$$q(\mathbf{z}_t|\mathbf{z}_s) = \mathcal{N}\left(\alpha_{t|s}\mathbf{z}_s, \sigma_{t|s}^2 \mathbf{I}\right), \quad \text{where} \quad \alpha_{t|s} = \alpha_t/\alpha_s, \quad \text{and} \quad \sigma_{t|s}^2 = \sigma_t^2 - \alpha_{t|s}^2 \sigma_s^2. \tag{2}$$

Given the distributions above, it is straightforward to verify that the reverse time inference distribution of $\mathbf{z}_s$ given $\mathbf{z}_t$ and $\mathbf{x}$, for any $0 \leq s < t \leq 1$, is also Gaussian and given by:

$$q(\mathbf{z}_s|\mathbf{z}_t, \mathbf{x}) = \mathcal{N}(\boldsymbol{\mu}_Q(\mathbf{z}_t, \mathbf{x}; s, t), \sigma_{Q,s,t}^2 \mathbf{I}) \quad \text{with} \quad \sigma_{Q,s,t}^2 = \sigma_{t|s}^2 \sigma_s^2/\sigma_t^2, \tag{3}$$

$$\text{and} \quad \boldsymbol{\mu}_Q(\mathbf{z}_t, \mathbf{x}; s, t) = \frac{1}{\alpha_{t|s}}(\mathbf{z}_t + \sigma_{t|s}^2 \nabla_{\mathbf{z}_t} \log q(\mathbf{z}_t|\mathbf{x})) = \frac{\alpha_{t|s}\sigma_s^2}{\sigma_t^2}\mathbf{z}_t + \frac{\alpha_s \sigma_{t|s}^2}{\sigma_t^2}\mathbf{x}. \tag{4}$$

In Appendix E we provide an implementation of $\sigma_{Q,s,t}^2$ that is numerically stable for small $t - s$.

As we show in Section 4, the continuous-time ELBO objective that we will propose optimizing is surprisingly invariant to the choice of functions α_t and σ_t, which we refer to as the *noise schedule*. Their only impact on our objective is through their ratio at times $t = 0$ and $t = 1$:

$$\text{SNR}(t) = \alpha_t^2/\sigma_t^2, \tag{5}$$

which we call the *signal-to-noise ratio*.

The specific diffusion processes used by Song and Ermon [2019] and Sohl-Dickstein et al. [2015] can be seen as special cases of the proposed model. Song and Ermon [2019] use $\alpha_t = 1$, called *variance-exploding* diffusion processes by Song et al. [2021]. In our experiments, we choose to use *variance-preserving* diffusion processes as in [Sohl-Dickstein et al., 2015, Ho et al., 2020] where $\alpha_t = \sqrt{1 - \sigma_t^2}$. Written as a function of $\text{SNR}(t)$, this is:

$$\alpha_t^2 = \text{SNR}(t)/(1 + \text{SNR}(t)) \quad \text{and} \quad \sigma_t^2 = 1/(1 + \text{SNR}(t)). \tag{6}$$

In previous works the signal-to-noise ratio $\text{SNR}(t)$ was a fixed function of time, but here we learn this function jointly with the rest of the model, as we explain in Section 5.

3.2 Reverse time generative model

We define our generative model by inverting the diffusion process of Section 3.1, yielding a hierarchical generative model that samples a sequence of latents $\mathbf{z}_t$, with time running backward from $t = 1$ to $t = 0$. We consider both the case where this sequence consists of a finite number of steps T, as well as a continuous time model corresponding to $T \to \infty$. We start by presenting the discrete-time case.

Given finite T, we discretize time uniformly into T segments of width $\tau = 1/T$. Defining $s(i) = (i - 1)/T$ and $t(i) = i/T$, our hierarchical generative model for data $\mathbf{x}$ is then given by:

$$p(\mathbf{x}) = \int_{\mathbf{z}} p(\mathbf{z}_1) p(\mathbf{x}|\mathbf{z}_0) \prod_{i=1}^{T} p(\mathbf{z}_{s(i)}|\mathbf{z}_{t(i)}). \tag{7}$$

With the variance preserving diffusion specification and sufficiently small $\text{SNR}(1)$, we have that $q(\mathbf{z}_1|\mathbf{x}) \approx \mathcal{N}(\mathbf{z}_1; 0, \mathbf{I})$. We therefore model the marginal distribution of $\mathbf{z}_1$ as a spherical Gaussian:

$$p(\mathbf{z}_1) = \mathcal{N}(\mathbf{z}_1; 0, \mathbf{I}) \tag{8}$$

For the reconstruction term, we wish to choose a model $p(\mathbf{x}|\mathbf{z}_0)$ that is close to the unknown $q(\mathbf{x}|\mathbf{z}_0)$. Let x_i and $z_{0,i}$ be the i-th elements of $\mathbf{x}, \mathbf{z}_0$, respectively. We then use a factorized distribution of the form:

$$p(\mathbf{x}|\mathbf{z}_0) = \prod_i p(x_i|z_{0,i}) \tag{9}$$

where we choose $p(x_i|z_{0,i}) \propto q(z_{0,i}|x_i)$. With sufficiently large SNR(0), this becomes a very close approximation to the true $q(\mathbf{x}|\mathbf{z}_0)$, as the influence of the unknown data distribution $q(\mathbf{x})$ is overwhelmed by the likelihood $q(\mathbf{z}_0|\mathbf{x})$.

Finally, we choose the conditional model distributions as

$$p(\mathbf{z}_s|\mathbf{z}_t) = q(\mathbf{z}_s|\mathbf{z}_t, \mathbf{x} = \hat{\mathbf{x}}_{\boldsymbol{\theta}}(\mathbf{z}_t; t)), \tag{10}$$

i.e. the same as the reverse time inference model $q(\mathbf{z}_s|\mathbf{z}_t, \mathbf{x})$, but with the original data $\mathbf{x}$ replaced by the output of a denoising model $\hat{\mathbf{x}}_{\boldsymbol{\theta}}(\mathbf{z}_t; t)$ that predicts $\mathbf{x}$ from its noisy version $\mathbf{z}_t$. We then have $p(\mathbf{z}_s|\mathbf{z}_t) = \mathcal{N}(\mathbf{z}_s; \boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{z}_t; s, t), \sigma^2_{Q,s,t}\mathbf{I})$, with

$$\boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{z}_t; s, t) = \frac{\alpha_{t|s}\sigma_s^2}{\sigma_t^2}\mathbf{z}_t + \frac{\alpha_s \sigma^2_{t|s}}{\sigma_t^2}\hat{\mathbf{x}}_{\boldsymbol{\theta}}(\mathbf{z}_t; t) = \frac{1}{\alpha_{t|s}}\mathbf{z}_t - \frac{\sigma^2_{t|s}}{\alpha_{t|s}\sigma_t}\hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(\mathbf{z}_t; t) = \frac{1}{\alpha_{t|s}}\mathbf{z}_t + \frac{\sigma^2_{t|s}}{\alpha_{t|s}}\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{z}_t; t), \tag{11}$$

with variance $\sigma^2_{Q,s,t}$ the same as in Equation 3, and with

$$\hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(\mathbf{z}_t; t) = (\mathbf{z}_t - \alpha_t \hat{\mathbf{x}}_{\boldsymbol{\theta}}(\mathbf{z}_t; t))/\sigma_t \quad \text{and} \quad \mathbf{s}_{\boldsymbol{\theta}}(\mathbf{z}_t; t) = (\alpha_t \hat{\mathbf{x}}_{\boldsymbol{\theta}}(\mathbf{z}_t; t) - \mathbf{z}_t)/\sigma_t^2. \tag{12}$$

Equation 11 shows that we can interpret our model in three different ways: 1) In terms of the *denoising model* $\hat{\mathbf{x}}_{\boldsymbol{\theta}}(\mathbf{z}_t; t)$ that recovers $\mathbf{x}$ from its corrupted version $\mathbf{z}_t$. 2) In terms of a *noise prediction model* $\hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(\mathbf{z}_t; t)$ that directly infers the noise $\boldsymbol{\epsilon}$ that was used to generate $\mathbf{z}_t$. 3) In terms of a *score model* $\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{z}_t; t)$, that at its optimum equals the scores of the marginal density: $\mathbf{s}^*(\mathbf{z}_t; t) = \nabla_{\mathbf{z}} \log q(\mathbf{z}_t)$; see Appendix G. These are three equally valid views on the same model class, that have been used interchangeably in the literature. We find the denoising interpretation the most intuitive, and will therefore mostly use $\hat{\mathbf{x}}_{\boldsymbol{\theta}}(\mathbf{z}_t; t)$ in this paper, although in practice we parameterize our model via $\hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(\mathbf{z}_t; t)$ following Ho et al. [2020].

When we take the number of steps $T \to \infty$, our model for $p(\mathbf{z}_t)$ can best be described as a continuous time diffusion process [Song et al., 2021], governed by the stochastic differential equation

$$d\mathbf{z}_t = [f(t)\mathbf{z}_t - g^2(t) s_{\boldsymbol{\theta}}(\mathbf{z}_t; t)]dt + g(t)d\mathbf{W}_t, \tag{13}$$

with time running backwards from $t = 1$ to $t = 0$ and

$$f(t) = \frac{d \log \alpha_t}{dt}, \qquad g^2(t) = \frac{d\sigma^2(t)}{dt} - 2\frac{d \log \alpha_t}{dt}\sigma^2(t). \tag{14}$$

As we argue in Section 4.2, we reach the best likelihoods with $T \to \infty$. For most practical purposes however, there is no difference between the continuous time formulation of our model and the discrete-time formulation with large T. For simplicity we therefore use the discrete-time model to derive most of our results in the remaining discussion. For a more detailed discussion of generative modeling via stochastic differential equations see Song et al. [2021].

4 The Evidence Lower Bound

Similar to the original DPMs [Sohl-Dickstein et al., 2015], we optimize the parameters towards the variational lower bound of the marginal likelihood, also called the evidence lower bound (ELBO). Unlike earlier DPMs, but similar to VAEs [Kingma and Welling, 2013], we optimize the inference model parameters (that define the forward time diffusion process) jointly with the rest of the model.

The negative marginal log-likelihood is bounded by:

$$-\log p(\mathbf{x}) \leq -\text{ELBO}(\mathbf{x}) = \underbrace{D_{KL}(q(\mathbf{z}_1|\mathbf{x})||p(\mathbf{z}_1))}_{\text{Prior loss}} + \underbrace{\mathbb{E}_{q(\mathbf{z}_0|\mathbf{x})}\left[-\log p(\mathbf{x}|\mathbf{z}_0)\right]}_{\text{Reconstruction loss}} + \underbrace{\mathcal{L}_T(\mathbf{x})}_{\text{Diffusion loss}} \tag{15}$$

The prior loss is a KL divergence between two Gaussians that can be computed in closed form; see Appendix B. The reconstruction loss can be evaluated and optimized using standard reparameterization gradients [Kingma and Welling, 2013]. The diffusion loss, $\mathcal{L}_T(\mathbf{x})$, is more complicated, and depends on the hyperparameter T that determines the depth of the model.

4.1 Discrete-time loss

In the case of finite T, using $s(i) = (i-1)/T$, $t(i) = i/T$, the diffusion loss is:

$$\mathcal{L}_T(\mathbf{x}) = \sum_{i=1}^{T} \mathbb{E}_{q(\mathbf{z}_{t(i)}|\mathbf{x})} D_{KL}[q(\mathbf{z}_{s(i)}|\mathbf{z}_{t(i)}, \mathbf{x})||p(\mathbf{z}_{s(i)}|\mathbf{z}_{t(i)})]. \tag{16}$$

In appendix B we show that this expression simplifies considerably, yielding:

$$\mathcal{L}_T(\mathbf{x}) = \mathbb{E}_{\boldsymbol{\epsilon}\sim\mathcal{N}(0,\mathbf{I}), i\sim U\{1,T\}} \left[\frac{T}{2} \left(\mathrm{SNR}(s) - \mathrm{SNR}(t)\right) ||\mathbf{x} - \hat{\mathbf{x}}_{\boldsymbol{\theta}}(\mathbf{z}_t; t)||_2^2 \right] \tag{17}$$

where $U\{1, T\}$ is the uniform distribution on the integers $\{1, \ldots, T\}$, and $\mathbf{z}_t = \alpha_t \mathbf{x} + \sigma_t \boldsymbol{\epsilon}$.

4.2 More steps is better

A natural question to ask is what the number of time segments T should be, and whether more segments is always better. In Appendix C we analyze the difference between the diffusion loss with T segments, $\mathcal{L}_T(\mathbf{x})$, and the diffusion loss with double that number of segments, $\mathcal{L}_{2T}(\mathbf{x})$, and find that

$$\mathcal{L}_{2T}(\mathbf{x}) - \mathcal{L}_T(\mathbf{x}) = \mathbb{E}_{t,\boldsymbol{\epsilon}} \left[c(t') \left(||\mathbf{x} - \hat{\mathbf{x}}_{\boldsymbol{\theta}}(\mathbf{z}_{t'}; t')||_2^2 - ||\mathbf{x} - \hat{\mathbf{x}}_{\boldsymbol{\theta}}(\mathbf{z}_t; t)||_2^2 \right) \right], \tag{18}$$

with $t' = t - 0.5T^{-1}$ and $c(t') = \mathrm{SNR}(t' - 0.5T^{-1}) - \mathrm{SNR}(t')$. Since $t' < t$, latent $\mathbf{z}_{t'}$ is a less noisy version of the data $\mathbf{x}$ from earlier in the diffusion process compared to $\mathbf{z}_t$, which means that predicting the uncorrupted data from $\mathbf{z}_{t'}$ is easier than from $\mathbf{z}_t$. If our trained model $\hat{\mathbf{x}}_{\boldsymbol{\theta}}$ is sufficiently good, we should thus always have that $\mathcal{L}_{2T}(\mathbf{x}) - \mathcal{L}_T(\mathbf{x}) < 0$, i.e. that our ELBO will always be better for a larger number of time segments.

4.3 Continuous-time loss

Since taking more time steps leads to a better ELBO, we take $T \to \infty$ in the remainder of this paper, effectively treating time t as continuous rather than discrete. In Appendix B we show that in this limit the diffusion loss $\mathcal{L}_T(\mathbf{x})$ simplifies further. Letting $\mathrm{SNR}'(t) = \frac{d\mathrm{SNR}(t)}{dt} = \frac{d\alpha_t^2/\sigma_t^2}{dt}$, we have:

$$\mathcal{L}_\infty(\mathbf{x}) = -\frac{1}{2}\mathbb{E}_{\boldsymbol{\epsilon}\sim\mathcal{N}(0,\mathbf{I})} \int_0^1 \mathrm{SNR}'(t) \, \|\mathbf{x} - \hat{\mathbf{x}}_{\boldsymbol{\theta}}(\mathbf{z}_t; t)\|_2^2 \, dt, \tag{19}$$

$$= -\frac{1}{2}\mathbb{E}_{\boldsymbol{\epsilon}\sim\mathcal{N}(0,\mathbf{I}), t\sim\mathcal{U}(0,1)} \left[\mathrm{SNR}'(t) \, \|\mathbf{x} - \hat{\mathbf{x}}_{\boldsymbol{\theta}}(\mathbf{z}_t; t)\|_2^2 \right]. \tag{20}$$

4.3.1 Invariance to the noise schedule in continuous time

Note that the signal-to-noise function $\mathrm{SNR}(t) = \alpha_t^2/\sigma_t^2$ is invertible due to the monotonicity assumption in Section 3.1. Due to this invertibility, we can perform a change of variables, and make everything a function of $v \equiv \mathrm{SNR}(t)$ instead of t, such that $t = \mathrm{SNR}^{-1}(v)$. Let α_v and σ_v be the functions α_t and σ_t evaluated at $t = \mathrm{SNR}^{-1}(v)$, and correspondingly let $\mathbf{z}_v = \alpha_v \mathbf{x} + \sigma_v \boldsymbol{\epsilon}$. Similarly, we rewrite our noise prediction model as $\tilde{\mathbf{x}}_{\boldsymbol{\theta}}(\mathbf{z}, v) \equiv \hat{\mathbf{x}}_{\boldsymbol{\theta}}(\mathbf{z}, \mathrm{SNR}^{-1}(v))$. With this change of variables, our continuous-time loss in Equation 19 can equivalently be written as:

$$\mathcal{L}_\infty(\mathbf{x}) = \frac{1}{2}\mathbb{E}_{\boldsymbol{\epsilon}\sim\mathcal{N}(0,\mathbf{I})} \int_{\mathrm{SNR}_{\mathrm{min}}}^{\mathrm{SNR}_{\mathrm{max}}} \|\mathbf{x} - \tilde{\mathbf{x}}_{\boldsymbol{\theta}}(\mathbf{z}_v, v)\|_2^2 \, dv, \tag{21}$$

where instead of integrating w.r.t. time t we now integrate w.r.t. the signal-to-noise ratio v, and where $\mathrm{SNR}_{\mathrm{min}} = \mathrm{SNR}(1)$ and $\mathrm{SNR}_{\mathrm{max}} = \mathrm{SNR}(0)$.

What this equation shows us is that the only effect the functions $\alpha(t)$ and $\sigma(t)$ have on the diffusion loss is through the values $\mathrm{SNR}(t) = \alpha_t^2/\sigma_t^2$ at endpoints $t = 0$ and $t = 1$. Given these values $\mathrm{SNR}_{\mathrm{max}}$ and $\mathrm{SNR}_{\mathrm{min}}$, the diffusion loss is invariant to the shape of function $\mathrm{SNR}(t)$ between $t = 0$ and $t = 1$.

4.3.2 Equivalence of diffusion models in continuous time

In the last section, we showed that the ELBO is only impacted by the function $\text{SNR}(t)$ through its endpoints $\text{SNR}_{\text{min}}, \text{SNR}_{\text{max}}$. We will now show that, apart from these endpoints, the choice of α_t and σ_t actually does not matter at all in continuous time: Since $v = \alpha_v^2/\sigma_v^2$, we have that $\sigma_v = \alpha_v/\sqrt{v}$, which means that $\mathbf{z}_v = \alpha_v \mathbf{x} + \sigma_v \boldsymbol{\epsilon} = \alpha_v(\mathbf{x} + \boldsymbol{\epsilon}/\sqrt{v})$. We can therefore adapt a model $\tilde{\mathbf{x}}_{\boldsymbol{\theta}}^A(\mathbf{z}, v)$ trained with diffusion process $\mathbf{z}_v^A = \alpha_v^A \mathbf{x} +, \sigma_v^A \boldsymbol{\epsilon}$, to a model $\tilde{\mathbf{x}}_{\boldsymbol{\theta}}^B$ corresponding to a different diffusion process $\mathbf{z}_v^B = \alpha_v^B \mathbf{x} +, \sigma_v^B \boldsymbol{\epsilon}$ simply by rescaling its input $\mathbf{z}$: Because $(\alpha_v^A/\alpha_v^B)\mathbf{z}_v^B = \mathbf{z}_v^A$, we can simply define $\tilde{\mathbf{x}}_{\boldsymbol{\theta}}^B(\mathbf{z}_v^B, v) \equiv \tilde{\mathbf{x}}_{\boldsymbol{\theta}}^A((\alpha_v^A/\alpha_v^B)\mathbf{z}_v^B, v)$. For $v' < v$, we now have that

$$
\begin{aligned}
&p_\theta^B(\mathbf{z}_{v'}^B = \mathbf{z} | \mathbf{z}_v^B) = q^B(\mathbf{z}_{v'}^B = \mathbf{z} | \mathbf{z}_v^B, \mathbf{x} = \tilde{\mathbf{x}}_{\boldsymbol{\theta}}^B(\mathbf{z}_v^B, v)) \\
&= q^A((\alpha_v^B/\alpha_v^A)\mathbf{z}_{v'}^A = \mathbf{z} | \mathbf{z}_v^A, \mathbf{x} = \tilde{\mathbf{x}}_{\boldsymbol{\theta}}^A(\mathbf{z}_v^A, v)) = p_\theta^A((\alpha_{v'}^B/\alpha_{v'}^A)\mathbf{z}_{v'}^A = \mathbf{z} | \mathbf{z}_v^A).
\end{aligned} \tag{22}
$$

In other words, diffusion processes A, B with their corresponding denoising models define the exact same generative model. Assuming these diffusion processes also start and stop at the same signal-to-noise ratios SNR_{max} and SNR_{min}, the previous section tells us that $\mathcal{L}_\infty^A(\tilde{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}^A, \mathbf{x}) = \mathcal{L}_\infty^B(\tilde{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}^B, \mathbf{x})$, i.e. they also both produce the same diffusion loss in continuous time. Any two diffusion models A and B, under the mild constraints set in 3.1 (which includes the variance exploding and variance preserving specifications), can thus be seen as equivalent in continuous time, up to a trivial re-scaling of $\mathbf{z}$.

This equivalence between diffusion specifications continues to hold even if, instead of the ELBO, these models optimize a *weighted* diffusion loss of the form:

$$
\mathcal{L}_\infty(\mathbf{x}, w) = \frac{1}{2}\mathbb{E}_{\boldsymbol{\epsilon}\sim(0,\mathbf{I})} \int_{\text{SNR}_{\text{min}}}^{\text{SNR}_{\text{max}}} w(v) \left\| \mathbf{x} - \tilde{\mathbf{x}}_{\boldsymbol{\theta}}(\mathbf{z}_v, v) \right\|_2^2 dv, \tag{23}
$$

which e.g. captures all the different objectives discussed by Song et al. [2021], see Appendix F. Here, $w(v)$ is a weighting function that generally puts increased emphasis on the noisier data compared to the ELBO, and which thereby can sometimes improve perceptual generation quality as measured by certain metrics like FID and Inception Score.

4.3.3 Practical stochastic approximation of the continuous-time loss

Since calculating the integral $\mathcal{L}_\infty(\mathbf{x})$; or its generalization $\mathcal{L}_\infty(\mathbf{x}, w)$, is not analytically tractable, we use its unbiased Monte Carlo estimator in practice:

$$
\mathcal{L}_\infty^{MC}(\mathbf{x}, w, \text{SNR}) = -\frac{1}{2}\text{SNR}'(t) w(\text{SNR}(t)) \left\| \mathbf{x} - \tilde{\mathbf{x}}_{\boldsymbol{\theta}}(\mathbf{z}_t; \text{SNR}(t)) \right\|_2^2, \tag{24}
$$

with $\mathbf{z}_t = \alpha_t \mathbf{x} + \sigma_t \boldsymbol{\epsilon}$, with $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I}), t \sim U[0, 1]$, and where we use $w(v) = 1$ as corresponding to the (unweighted) ELBO. The resulting ELBO estimate can then be optimized using stochastic gradient descent as usual. The noise schedule $\text{SNR}(t)$ influences the variance of our Monte Carlo estimator, which is why we jointly optimize it with the rest of the model as described in Section 5.

Here, we also found it helpful to sample time t using a low-discrepancy sampler. When processing a minibatch of k examples $\mathbf{x}^i$, $i \in \{1, \ldots, k\}$, we require k timesteps t^i sampled from a uniform distribution. Instead of sampling these timesteps independently, we sample a single uniform random number $u_0 \sim U[0, 1]$ and then set $t^i = \text{mod}(u_0 + i/k, 1)$. Each t^i now as the correct uniform marginal distribution, but the minibatch of timesteps covers the space in $[0, 1]$ more equally than when sampling independently, which we find to reduce the variance in our ELBO estimate.

5 Learning the noise schedule

So far, we have not discussed how to select the signal-to-noise ratio function, or *noise schedule* $\text{SNR}(t) = \alpha_t^2/\sigma_t^2$ that governs our generative model. In previous work, $\text{SNR}(t)$ has a fixed form (see Appendix D, Fig. 4a). Here, we propose learning this schedule jointly with our denoising model $\hat{\mathbf{x}}_{\boldsymbol{\theta}}$. We parameterize $\text{SNR}(t) = \exp(-\gamma_{\boldsymbol{\eta}}(t))$ using a monotonically increasing neural network $\gamma_{\boldsymbol{\eta}}(t)$, details of which are given in Appendix D.

In the discrete-time case, we then learn parameters $\boldsymbol{\eta}$ by maximizing the ELBO, together with our other model parameters. We find this to be especially beneficial when T is small, as we show in Section 7.

The continuous-time case is different: As we showed in Section 4.3.1, the continuous-time diffusion loss is invariant to $\mathrm{SNR}(t)$, except for its endpoints $\mathrm{SNR}_{\text{min}} = \alpha_1^2/\sigma_1^2$ and $\mathrm{SNR}_{\text{max}} = \alpha_0^2/\sigma_0^2$. For this case, we therefore only optimize the ELBO with respect to $\mathrm{SNR}_{\text{min}}, \mathrm{SNR}_{\text{max}}$, and not the parameters $\boldsymbol{\eta}$ of the schedule interpolating between them.

Although this interpolating function $\gamma_{\boldsymbol{\eta}}$ does not impact the value of the continuous-time diffusion loss, it does impact the variance of our stochastic estimate of it, given in Equation 24. We therefore propose to learn $\boldsymbol{\eta}$ by minimizing the variance, which we do by performing stochastic gradient descent on our squared diffusion loss $\mathcal{L}_\infty^{MC}(\mathbf{x}, w, \gamma)^2$. We have that $\mathbb{E}[\mathcal{L}_\infty^{MC}(\mathbf{x}, w, \gamma)^2] = \mathcal{L}_\infty(\mathbf{x}, w)^2 + \mathrm{Var}[\mathcal{L}_\infty^{MC}(\mathbf{x}, w, \gamma)]$, where the first part is independent of $\gamma_{\boldsymbol{\eta}}(t)$, and hence that

$$\mathbb{E}[\nabla_{\boldsymbol{\eta}} \mathcal{L}_\infty^{MC}(\mathbf{x}, w, \gamma_{\boldsymbol{\eta}})^2] = \nabla_{\boldsymbol{\eta}} \mathrm{Var}[\mathcal{L}_\infty^{MC}(\mathbf{x}, w, \gamma_{\boldsymbol{\eta}})]. \tag{25}$$

We can calculate this gradient with negligible computational overhead as a by-product of calculating the gradient of the ELBO, details of which are given in Appendix D.

This strategy of minimizing the variance of our diffusion loss estimate remains valid for weighted diffusion losses, $w(v) \neq 1$, not corresponding to the ELBO, and we therefore expect it to be useful beyond the goal of optimizing for likelihood that we consider in this paper.

6 Fourier features for improved fine scale prediction

Prior work on diffusion models has mainly focused on the perceptual quality of generated samples, which emphasizes coarse scale patterns and global consistency of generated images. Here, we optimize for likelihood, which is sensitive to fine scale details and exact values of individual pixels. Since our reconstruction model $p(\mathbf{x}|\mathbf{z}_0)$ given in Equation 9 is weak, the burden of modeling these fine scale details falls on our denoising diffusion model $\hat{\mathbf{x}}_{\boldsymbol{\theta}}$. In initial experiments, we found that the denoising model had a hard time accurately modeling these details. At larger noise levels, the latents $\mathbf{z}_t$ follow a smooth distribution due to the added Gaussian noise, but at the smallest noise levels the discrete nature of 8-bit image data leads to sharply peaked marginal distributions $q(\mathbf{z}_t)$.

To capture the fine scale details of the data, we propose adding a set of *Fourier features* to the input of our denoising model $\hat{\mathbf{x}}_{\boldsymbol{\theta}}(\mathbf{z}_t; t)$. Such Fourier features consist of a linear projection of the original data onto a set of periodic basis functions with high frequency, which allows the network to more easily model high frequency details of the data. Previous work [Tancik et al., 2020] has used these features for input coordinates to model high frequency details across the *spatial* dimension, and for time embeddings to condition denoising networks over the *temporal* dimension[Song et al., 2021]. Here we apply it to color channels for single pixels, in order to model fine distributional details at the level of each scalar input.

Concretely, let $z_{i,j,k}$ be the scalar value in the k-th channel in the (i, j) spatial position of network input $\mathbf{z}_t$. We then add four additional channels to the input of the denoising model of the form

$$f_{i,j,k}^q = \sin(z_{i,j,k} 2^q \pi), \quad \text{and} \quad g_{i,j,k}^q = \cos(z_{i,j,k} 2^q \pi), \quad \text{for} \quad q \in \{7, 8\}. \tag{26}$$

These four additional channels are then concatenated to $\mathbf{z}_t$ before being used as input in a standard convolutional denoising model similar to that used by Ho et al. [2020]. We find that the presence of these high frequency features allows our network to learn with much higher values of $\mathrm{SNR}_{\text{max}}$, or conversely lower noise levels σ_0^2, than is otherwise optimal. This leads to large improvements in likelihood as demonstrated in Section 7 and Figure 3. We did not observe such improvements when incorporating Fourier features into autoregressive models.

7 Experiments

We demonstrate our proposed class of diffusion models, which we call *Fusion*, on the CIFAR-10 [Krizhevsky et al., 2009] and downsampled ImageNet [Van Oord et al., 2016, Deng et al., 2009] datasets, where we focus on maximizing likelihood. The score models we use closely follow Ho et al. [2020], except that they process the data solely at the original resolution, without any internal downsampling or upsampling. Our score models are also deeper than the diffusion models used by others in the literature. All reported models incorporate Fourier features (Section 6) as well as a learnable diffusion specification (Section 5). Complete details on our model specifications can be found in Appendix A.

Model (Bits per dimension on test set)	Type	CIFAR10	ImageNet 32x32	ImageNet 64x64
ResNet VAE with IAF [Kingma et al., 2016]	VAE	3.11		
Very Deep VAE [Child, 2020]	VAE	2.87	3.80	3.52
Glow [Kingma and Dhariwal, 2018]	Flow	3.35	4.09	3.81
Flow++ [Ho et al., 2019]	Flow	3.08	3.86	3.69
PixelCNN [Van Oord et al., 2016]	AR	3.03	3.83	3.57
PixelCNN++ [Salimans et al., 2017]	AR	2.92		
Image Transformer [Parmar et al., 2018]	AR	2.90	3.77	
SPN 150M [Menick and Kalchbrenner, 2018]	AR			3.52
Sparse Transformer [Child et al., 2019]	AR	2.80		3.44
Routing Transformer [Roy et al., 2021]	AR			3.43
DDPM [Ho et al., 2020]	Diff	3.69		
Score SDE [Song et al., 2021]	Diff	2.99		
Improved DDPM [Nichol and Dhariwal, 2021]	Diff	2.94		3.54
Fusion (ours)	Diff	**2.66**	**3.72**	**3.40**

Table 1: Summary of our findings for density modeling tasks, in terms of bits per dimension (BPD) on the test set. We only list methods without data augmentation. Model types are autoregressive (AR), normalizing flows (Flow), variational autoencoders (VAE), or diffusion models (Diff). Numbers for VAEs and diffusion models are upper bounds. Our results were obtained using the continuous-time formulation of our model.

Table 1 shows our results on modeling the CIFAR-10 and downsampled ImageNet datasets. We establish a new state-of-the-art in terms of test set likelihood, by a significant margin. Our CIFAR-10 model surpasses the previous state-of-the-art about 10x faster than it takes the Sparse Transformer to reach this, in wall clock time on equivalent hardware.

Our CIFAR-10 model, whose hyper-parameters were tuned for likelihood, results in a FID (perceptual quality) score of 7.41. This would have been state-of-the-art until recently, but is worse than recent diffusion models that specifically target FID scores [Nichol and Dhariwal, 2021, Song et al., 2021, Ho et al., 2020]. We found that our method can get significantly better FIDs when similarly tuning for FID instead of likelihood, but we did not pursue this direction here.

A random sample of generated images from our model is provided in Figure 2. We provide additional samples from this model, as well as our other models for the other datasets, in Appendix I.

Figure 2: Non cherry-picked unconditional samples from our Imagenet 64x64 model, trained in continuous time and generated using $T = 1000$.

Next, we investigate the relative importance of our contributions. In Table 2 we compare our discrete-time and continuous-time specifications of the diffusion model: When evaluating our model with a small number of steps, our discretely trained models perform better by learning the diffusion schedule to optimize the ELBO. However, as argued theoretically in Section 4.2, we find experimentally that more steps T indeed gives better likelihood. When T grows large, our continuously trained model performs best, helped by training its diffusion schedule to minimize variance instead.

Minimizing the variance also helps the continuous time model to train faster, as shown in Figure 3. This effect is further examined in Table 4b, where we find dramatic variance reductions compared to

our baselines in continuous time. Figure 4a shows how this effect is achieved: Compared to the other schedules, our learned schedule spends much more time in the high SNR(t) / low σ_t^2 range.

T_{train}	T_{eval}	BPD
10	10	4.31
100	100	2.84
250	250	2.74
500	500	2.71
1000	1000	2.69
10000	10000	2.68
∞	100	3.63
∞	1000	2.75
∞	10000	2.68
∞	∞	**2.66**

Table 2: Discrete versus continuous-time training and evaluation with CIFAR-10, in terms of bits per dimension (BPD).

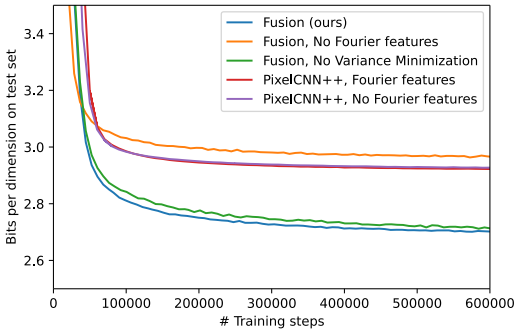
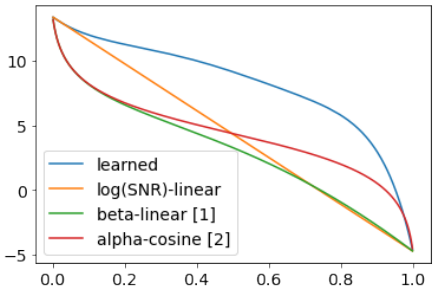


Figure 3: Test set likelihoods during training, with/without Fourier features, and with/without learning the noise schedule to minimize variance.



(a) log SNR vs time t

SNR(t) schedule	Var(BPD)
Learned (ours)	**0.53**
log SNR-linear	6.35
β-Linear [1]	31.6
α-Cosine [2]	31.1

(b) Variance of ELBO estimate

Figure 4: Our learned continuous-time variance-minimizing noise schedule SNR(t) for CIFAR-10, compared to its log-linear initialization and to schedules from the literature: [1] The β-Linear schedule from Ho et al. [2020], [2] The α-Cosine schedule from Nichol and Dhariwal [2021]. All schedules were scaled and shifted on the log scale such that the resulting $\text{SNR}_{\text{min}}, \text{SNR}_{\text{max}}$ were the equal to our learned endpoints, resulting in the same ELBO estimate of 2.66. We report the variance of our ELBO estimate per data point, computed on the test set, and conditional on the data: This does not include the noise due to sampling minibatches of data.

In Figure 3 we further show training curves for our model including and excluding the Fourier features proposed in Section 6: With Fourier features enabled our model achieves much better likelihood. For comparison we also implemented Fourier features in a PixelCNN++ model [Salimans et al., 2017], where we do not see a benefit.

8 Conclusion

We presented state-of-the-art results on modeling the density of natural images using a new class of diffusion models that incorporates a learnable diffusion specification, Fourier features for fine-scale modeling, as well as other architectural innovations. In addition, we obtained new theoretical insight into likelihood-based generative modeling with diffusion models, showing a surprising invariance of the ELBO to the forward time diffusion process in continuous time, as well as an equivalence between various diffusion processes from the literature previously thought to be different.

Although our focus is on improving likelihood, the proposed methods can potentially be used to improve perceptual quality metrics like FID and Inception Score, potentially impacting applications with societal impact such as fake content generation. However, we expect a greater immediate impact on likelihood-based applications such as lossless compression, for which we see a lower risk of negative impact.

References

Nanxin Chen, Yu Zhang, Heiga Zen, Ron J. Weiss, Mohammad Norouzi, and William Chan. Wavegrad: Estimating gradients for waveform generation. *arXiv preprint arXiv:2009.00713*, 2020.

Rewon Child. Very deep VAEs generalize autoregressive models and can outperform them on images. *arXiv preprint arXiv:2011.10650*, 2020.

Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*, 2019.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

Conor Durkan and Yang Song. On maximum likelihood training of score-based generative models. *arXiv preprint arXiv:2101.09258*, 2021.

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.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *arXiv preprint arXiv:2006.11239*, 2020.

Diederik P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. *arXiv preprint arXiv:1807.03039*, 2018.

Diederik P Kingma and Max Welling. Auto-encoding variational Bayes. *International Conference on Learning Representations*, 2013.

Diederik P Kingma, Tim Salimans, Rafal Jozefowicz, Xi Chen, Ilya Sutskever, and Max Welling. Improved variational inference with inverse autoregressive flow. In *Advances in Neural Information Processing Systems*, pages 4743–4751, 2016.

Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. Diffwave: A versatile diffusion model for audio synthesis. *arXiv preprint arXiv:2009.09761*, 2020.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

David JC MacKay. *Information theory, inference and learning algorithms*. Cambridge university press, 2003.

Jacob Menick and Nal Kalchbrenner. Generating high fidelity images with subscale pixel networks and multidimensional upscaling. *arXiv preprint arXiv:1812.01608*, 2018.

Alex Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. *arXiv preprint arXiv:2102.09672*, 2021.

Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016.

Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Lukasz Kaiser, Noam Shazeer, Alexander Ku, and Dustin Tran. Image transformer. In *International Conference on Machine Learning*, pages 4055–4064. PMLR, 2018.

Danilo J Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *International Conference on Machine Learning*, pages 1278–1286, 2014.

Aurko Roy, Mohammad Saffar, Ashish Vaswani, and David Grangier. Efficient content-based sparse attention with routing transformers. *Transactions of the Association for Computational Linguistics*, 9:53–68, 2021.

Tim Salimans, Andrej Karpathy, Xi Chen, and Diederik P Kingma. Pixelcnn++: Improving the pixelcnn with discretized logistic mixture likelihood and other modifications. *arXiv preprint arXiv:1701.05517*, 2017.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pages 2256–2265, 2015.

Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. In *Advances in Neural Information Processing Systems*, pages 11895–11907, 2019.

Yang Song and Stefano Ermon. Improved techniques for training score-based generative models. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 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*, 2021.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. *arXiv preprint arXiv:1409.3215*, 2014.

Matthew Tancik, Pratul P Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan T Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. *arXiv preprint arXiv:2006.10739*, 2020.

Aaron Van Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. In *International Conference on Machine Learning*, pages 1747–1756, 2016.

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

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] Limitations are discussed in Section 8
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] Section 8 mentions lack of improvement on sample quality may restrict harm in applications such as deepfakes
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes] Proofs for all theoretical results are in the Appendix
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [No] We have not included the code, but intend to open source it before the conference.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] Training details are in Appendix A
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No] We do not include error bars due to the computational demands of individual experiments, but empirically we have observed that variation across seeds is small.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Appendix A
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [No] These are standard benchmark datasets
   (c) Did you include any new assets either in the supplemental material or as a URL? [No]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [No]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [No]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]