BDDM: BILATERAL DENOISING DIFFUSION MODELS FOR FAST AND HIGH-QUALITY SPEECH SYNTHESIS

Anonymous authors
Paper under double-blind review

ABSTRACT

Denoising diffusion probabilistic models (DDPMs) have emerged as competitive generative models yet brought challenges to efficient sampling. In this paper, we propose a novel bilateral denoising diffusion model (BDDM), which takes significantly fewer steps to generate high-quality audio samples. From a bilateral modeling objective, BDDMs parameterize the forward and reverse processes with a score network and a scheduling network, respectively. We show that a new surrogate objective can be derived for training the two networks. For practical concerns, BDDMs are efficient, simple-to-train, and capable of further improving any pre-trained DDPM by optimizing the inference noise schedules. Our experiments demonstrated that BDDMs can generate high-fidelity audio samples with as few as 3 inference steps and produce comparable or even higher quality samples than the state-of-the-art diffusion-based neural vocoder, with only 7 sampling steps (143x faster than WaveGrad and 28.6x faster than DiffWave).

1 INTRODUCTION

Deep generative models have shown a tremendous advancement in speech synthesis (van den Oord et al., 2016; Kalchbrenner et al., 2018; Prenger et al., 2019; Kumar et al., 2019a; Kong et al., 2020b; Chen et al., 2020; Kong et al., 2021). The successful generative models can be mainly divided into two branches – the GAN (Goodfellow et al., 2014) based models and the likelihood-based models. The former use an adversarial training procedure, but the training is usually unstable, and the relevant training objectives are not suitable to compare different GAN models; the latter use log-likelihood or surrogate objectives for training, but they also have intrinsic limitations. For example, the autoregressive models (van den Oord et al., 2016; Kalchbrenner et al., 2018) have exceedingly slow sampling speed and poor scaling properties on high dimensional data, as its sampling is inherently a sequential process. Likewise, the flow-based models (Dinh et al., 2016; Kingma & Dhariwal, 2018; Chen et al., 2018; Papamakarios et al., 2021) rely on specialized architectures to build a normalized probability model so that the model training is less parameter-efficient and typically more difficult than other likelihood-based generative models (Bond-Taylor et al., 2021). On the other hand, for those using the surrogate objectives, such as the evidence lower bound in variational auto-encoders (Kingma & Welling, 2013; Rezende et al., 2014; Maaløe et al., 2019) and the contrastive divergence in energy-based models (Hinton, 2002; Carreira-Perpinan & Hinton, 2005), they improve inference speed but often yield lower-quality samples than autoregressive and GAN-based models and typically only work well for low-dimensional data (Bond-Taylor et al., 2021).

An up-and-coming class of likelihood-based models is the diffusion probabilistic models (DPMs) (Ho et al., 2020; Sohl-Dickstein et al., 2015), which shed light on the generation of high-quality samples comparable or even superior (Dhariwal & Nichol, 2021) to the current state-of-the-art (SOTA) autoregressive and GAN models. Sohl-Dickstein et al. (Sohl-Dickstein et al., 2015) introduced the idea of using an iterative forward process for destroying the structure of a given distribution while learning the reverse process for restoring the structure in the data. With a different modeling objective, the score-based generative models (Song & Ermon, 2019) used a neural network trained with the score matching (Hyvarinen & Dayan, 2005) to produce samples via Langevin dynamics. As a seminal work of DPMs, Ho et al. (Ho et al., 2020) proposed the denoising diffusion probabilistic models (DDPMs), which exploited a connection (Vincent, 2011) between the above two ideas that the diffusion framework can be viewed as a specific approach to provide support to score matching.

For the first time, DDPMs present high-quality image synthesis results on par with GAN models, especially on several higher-resolution image benchmarks. Subsequently, score matching and denoising diffusion models were applied to speech syntheses, such as Wavegrad (Chen et al., 2020) and a parallel contribution Diffwave (Kong et al., 2021), and demonstrated the capability of generating high fidelity audio samples outperforming non-autoregressive GAN-based models (Yamamoto et al., 2020; Kumar et al., 2019b; Yang et al., 2020; Bińkowski et al., 2020) meanwhile matching the quality of SOTA autoregressive methods (Chen et al., 2020).

Despite the compelling results, the diffusion models are two to three orders of magnitude slower than other generative models such as GANs and VAEs. The major limitation of the generative diffusion models is that it requires up to thousands of diffusion steps during training to learn the target distribution; Therefore, a large number of denoising steps are often required at sampling time. Recently, extensive investigations have been conducted to accelerate the sampling process for efficiently generating high-quality outputs. In WaveGrad (Chen et al., 2020), a grid search algorithm was used to generate high-fidelity audio samples with six steps. From a different aspect, Song et al. (Song et al., 2021b) used a neural probability flow ODE to enable a fast deterministic multi-step sampling process. A parallel contribution called the denoising diffusion implicit models (DDIMs) (Song et al., 2021a) considered non-Markovian diffusion processes and used a subsequence of the noise schedule to accelerate the denoising process. A more recent work (Luhman & Luhman, 2021) explored a student-teacher method to distill the DDIMs sampling process into a single-step model.

Previous diffusion models considered alternative forms of reverse process or using additional knowledge for better conditional generation. Distinctively, we noticed that the reduction of the sampling steps essentially depended on the choice of a noise schedule, which was conventionally considered yet as predetermined for the forward process. We are thus motivated to learn the noise schedule within the network training framework directly. With such an incentive, we propose the bilateral denoising diffusion models (BDDM), which parameterize the forward and reverse processes, with a score network and a scheduling network, respectively. There are several nice properties of BDDMs. First, in BDDMs, we prove that a new lower bound for the log marginal likelihood tighter than the conventional ELBO can be derived. Secondly, we can also derive new objectives for learning both the score network and the scheduling network. Interestingly, our derived loss for the score network resembles the objective used in DDPMs under a reasonable condition. Thirdly, BDDMs allow for efficient training that merely adds a fraction of the original DDPM's score network training time. In particular, BDDM is efficient, simple to train, and can also be applied to any pre-trained DDPM for optimizing the noise schedules. Our experiments demonstrated that BDDMs can generate high-fidelity samples with as few as 3 sampling steps, and produce even higher quality of samples than the SOTA WaveGrad using 1000 steps with only 7 sampling steps (a 143x speedup).

2 BACKGROUND

2.1 DIFFUSION PROBABILISTIC MODELS (DPMS)

Given i.i.d. samples $\{\boldsymbol{x}_0 \in \mathbb{R}^D\}$ from an unknown data distribution $p_{\text{data}}(\boldsymbol{x}_0)$, diffusion probabilistic models (DPMs) (Sohl-Dickstein et al., 2015) define a forward process $q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0) = \prod_{t=1}^{T} q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$ that converts any complex data distribution into a simple, tractable distribution after T steps of diffusion. To revert the forward process, a reverse process $p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)$ parameterized by θ is used to model the data distribution: $p_\theta(\boldsymbol{x}_0) = \int \pi(\boldsymbol{x}_T) \prod_{t=1}^{T} p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t) d\boldsymbol{x}_{1:T}$, where $\pi(\boldsymbol{x}_T)$ is the prior distribution for starting the reverse process. Then, the variational parameters θ can be learnt by maximizing the standard evidence lower bound (ELBO):

$$\mathcal{F}_{\text{elbo}} := \mathbb{E}_q \left[\log p_\theta(\boldsymbol{x}_0|\boldsymbol{x}_1) - \sum_{t=2}^{T} \mathbb{KL}\left(q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0) || p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)\right) - \mathbb{KL}\left(q(\boldsymbol{x}_T|\boldsymbol{x}_0) || \pi(\boldsymbol{x}_T)\right) \right]. \tag{1}$$

2.2 DENOISING DIFFUSION PROBABILISTIC MODELS (DDPMS)

As an extension to DPMs, de-noising diffusion probabilistic models (DDPMs) (Ho et al., 2020) applied the score matching technique (Hyvarinen & Dayan, 2005; Song & Ermon, 2019) to define

the reverse process. In particular, DDPMs considered a Gaussian diffusion process parameterized by a *noise schedule* $\boldsymbol{\beta} \in \mathbb{R}^T$ with $0 < \beta_1, \ldots, \beta_T < 1$:

$$q_{\boldsymbol{\beta}}(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0) := \prod_{t=1}^{T} q_{\beta_t}(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}), \quad \text{where} \quad q_{\beta_t}(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) := \mathcal{N}(\sqrt{1-\beta_t}\boldsymbol{x}_{t-1}, \beta_t \boldsymbol{I}). \tag{2}$$

Based on the nice property of isotropic Gaussians, we can directly express $\boldsymbol{x}_t$ in terms of $\boldsymbol{x}_0$:

$$\boldsymbol{x}_t = \alpha_t \boldsymbol{x}_0 + \left(1-\alpha_t^2\right)\boldsymbol{\epsilon}, \quad \text{where} \quad \alpha_t = \prod_{i=1}^{t} \sqrt{1-\beta_i} \quad \text{and} \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}). \tag{3}$$

To revert this forward process, DDPMs employ a score network[1] $\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, \alpha_t)$ to define

$$p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t) := \mathcal{N}\left(\frac{1}{\sqrt{1-\beta_t}}\left(\boldsymbol{x}_t - \frac{\beta_t}{\sqrt{1-\alpha_t^2}}\boldsymbol{\epsilon}_\theta\left(\boldsymbol{x}_t, \alpha_t\right)\right), \Sigma_t\right), \tag{4}$$

where Σ_t is the co-variance matrix defined for the reverse process. Ho et al. (2020) showed that setting $\Sigma_t = \frac{1-\alpha_{t-1}^2}{1-\alpha_t^2}\beta_t \boldsymbol{I}$ is optimal for a deterministic $\boldsymbol{x}_0$, while setting $\Sigma_t = \beta_t \boldsymbol{I}$ is optimal for a white noise $\boldsymbol{x}_0 \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$. Nichol & Dhariwal (2021) also proposed to learn Σ_t jointly with the score network by doubling the number of output channels. Note that using the complete ELBO in Eq. (1) to train the score network requires T forward passes of the score network and back-propagating through T diffusion steps, which would make the training computationally prohibitive. For feasibly training the score network, instead of computing the complete ELBO, DDPMs proposed an efficient training approach by sampling a $t \sim \text{Uniform}(\{1, ..., T\})$ to compute a loss at each step of training:

$$\mathcal{L}_{\text{ddpm}}^{(t)}(\theta) := \left\| \boldsymbol{\epsilon}_t - \boldsymbol{\epsilon}_\theta\left(\alpha_t \boldsymbol{x}_0 + \sqrt{1-\alpha_t^2}\boldsymbol{\epsilon}_t, \alpha_t\right)\right\|_2^2, \tag{5}$$

which can be derived as a re-weighted term of $\mathbb{KL}\left(q_{\boldsymbol{\beta}}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0)||p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)\right)$. Although the re-weighting effectively worked in practice, it is deficient for learning the noise schedule $\boldsymbol{\beta}$ as shown in the ablation experiment in the Appendix. Distinctively, in the following sections, we derive an alternative noise scheduling approach from a bilateral modeling perspective that can generate comparable or even higher quality of audio samples with significantly fewer sampling steps.

3 BILATERAL DENOISING DIFFUSION MODELS (BDDMS)

Noise scheduling has been shown significant (Chen et al., 2020; Kong et al., 2021; Kong & Ping, 2021) for sampling acceleration (Chen et al., 2020) and for high-fidelity generation. In DDPMs, a linear noise schedule was set and shared for both forward and reverse processes (Ho et al., 2019). In practice, a large T, such as $T = 1000$, needs to be used to make the reverse process a good approximation. This leads to an unacceptably slow sampling process due to T forward passes of the score network according to Eq. (4).

To relax the computational burden, Chen et al. (Chen et al., 2020) and Kong et al. (Kong et al., 2021) allowed the noise schedule for sampling, denoted as $\hat{\boldsymbol{\beta}} \in \mathbb{R}^N$, to be different from the noise schedule for training $\boldsymbol{\beta}$. In particular, much shorter, non-linear noise schedules (as few as 6 steps) were successfully used in both works for sampling without observing any significant degradation in generation quality. However, searching for a highly-performed noise schedule for sampling remains an unsolved issue. Kong et al. (Kong et al., 2021) defined a fixed, practically useful noise schedule for sampling. As a more general approach, given a well-trained score network, Chen et al. (Chen et al., 2020) used a grid search (GS) algorithm to select $\hat{\boldsymbol{\beta}}$. Yet, GS is prohibitively slow for $N > 6$. Meanwhile, DDPMs (Ho et al., 2019) mentioned that the training noise schedule $\boldsymbol{\beta}$ could be learned by reparameterization (although not implemented or investigated in their work). We present an ablation study in Sec. 5 and demonstrate that a direct reparameterization on $\boldsymbol{\beta}$ is less favorable.

[1]Here, $\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, \alpha_t)$ is conditioned on the continuous noise scale α_t, as in (Song et al., 2021b; Chen et al., 2020). Alternatively, the score network can also be conditioned on a discrete time index $\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t)$, as in (Song et al., 2021a; Ho et al., 2020). An approximate mapping of a noise schedule to a time schedule (Kong & Ping, 2021) exists, we consider conditioning on noise scales as the general case.

In this paper, bilateral denoising diffusion models (BDDMs) are motivated (1) to provide a theoretical grounding on the training performed at the diffusion step t, which is inline with the training algorithm in (Ho et al., 2020), and (2) to allow a direct learning of the noise schedule for sampling based on the theory. The idea underlying BDDMs is to use the forward process to directly sample $\boldsymbol{x}_t \sim q_{\boldsymbol{\beta}}(\boldsymbol{x}_t|\boldsymbol{x}_0)$ and then consider the reverse process $p_\theta(\boldsymbol{x}_{0:t-1}|\boldsymbol{x}_t)$ starting from the sampled $\boldsymbol{x}_t$. The joint consideration of forward process and the reverse process in one integral is what we view as the bilateral modeling in BDDMs. Formally, we first define the prior at step t as $\pi(\boldsymbol{x}_t)$ such that $p_\theta(\boldsymbol{x}_{0:t}) = \pi(\boldsymbol{x}_t)p_\theta(\boldsymbol{x}_{0:t-1}|\boldsymbol{x}_t)$ for $t \in \{2, ..., T\}$. Then, we derive a new lower bound to the log marginal likelihood:

Proposition 1. *Given a noise schedule $\boldsymbol{\beta}$, the following lower bound holds for $t \in \{2, ..., T\}$:*

$$\log p_\theta(\boldsymbol{x}_0) \geq \mathcal{F}^{(t)}_{score}(\theta) := -\mathbb{E}_{q_{\boldsymbol{\beta}}(\boldsymbol{x}_t|\boldsymbol{x}_0)}\left[\mathcal{L}^{(t)}_{score}(\theta) + \mathcal{R}_\theta(\boldsymbol{x}_0, \boldsymbol{x}_t)\right], \tag{6}$$

where

$$\mathcal{L}^{(t)}_{score}(\theta) := \mathbb{KL}\left(p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)||\pi(\boldsymbol{x}_{t-1})\right), \tag{7}$$

$$\mathcal{R}_\theta(\boldsymbol{x}_0, \boldsymbol{x}_t) := -\mathbb{E}_{p_\theta(\boldsymbol{x}_1|\boldsymbol{x}_t)}\left[\log p_\theta(\boldsymbol{x}_0|\boldsymbol{x}_1)\right]. \tag{8}$$

The proof is provided in the Appendix. In practice, $\mathcal{R}_\theta(\boldsymbol{x}_0, \boldsymbol{x}_t)$ can be approximated by a one-step prediction from $\boldsymbol{x}_t$ to $\boldsymbol{x}_1$ using the non-Markovian probability density function defined in denoising diffusion implicit models (DDIMs) (Song et al., 2021a). In this regard, the proposed lower bound $\mathcal{F}^{(t)}_{\text{score}}(\theta)$ allows us to consider only one t at each training step for efficient training, which is practically more advantageous than the standard ELBO in Eq. (1) that entails computing a sum of T KL terms. Below, we show that the proposed bound resembles $\mathcal{L}^{(t)}_{\text{ddpm}}(\theta)$ under the following conditions:

Proposition 2. *If we set $\pi(\boldsymbol{x}_{t-1}) = q_{\boldsymbol{\beta}}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0)$ for $t \in \{2, ..., T\}$, then any optimal solution satisfying $\theta^* = argmin_\theta \mathcal{L}^{(t)}_{ddpm}(\theta)\, \forall t \in \{1, ..., T\}$, also satisfies $\theta^* = argmax_\theta \mathcal{F}^{(t)}_{score}(\theta)\, \forall t \in \{2, ..., T\}$.*

The proof is followed in the Appendix. By this proposition, we see that optimizing $\mathcal{L}^{(t)}_{\text{ddpm}}(\theta)$ for training the score network θ is sufficient to maximize the proposed lower bound $\mathcal{F}^{(t)}_{\text{score}}(\theta)$. In this sense, we provide an alternative theoretical explanation for the training algorithm used in DDPMs and DDIMs, which can be seen as optimizing a special case of our derived objective (when setting $\pi(\boldsymbol{x}_{t-1}) = q_{\boldsymbol{\beta}}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0)$). In addition, it is practically beneficial that our proposed BDDMs can re-use θ from any well-trained DDPM or DDIM.

Given that θ can be trained to maximize the log evidence with the pre-specified noise schedule $\boldsymbol{\beta}$, the remaining question of interest in BDDMs is how to find a good enough noise schedule $\hat{\boldsymbol{\beta}} \in \mathbb{R}^N$ for sampling for an optimized θ^*. To tackle this problem, we introduce a novel *scheduling network* $\sigma_\phi(\boldsymbol{x}_n)$, which is responsible for estimating the ratio between two consecutive noise scales ($\hat{\beta}_n$ and $\hat{\beta}_{n+1}$) from the current noisy observation $\boldsymbol{x}_n$ for inference in a descending order (from $n = N$ to 1). To make use of the scheduling, we first need to understand the relationship between the noise scale $\hat{\beta}_n$ at step n and the noise scale variables at step $n+1$ (i.e. $\hat{\alpha}_{n+1}$ and $\hat{\beta}_{n+1}$). As a remark, we indeed can derive the range of $\hat{\beta}_n$ from its definition:

Remark 1. *Suppose the noise schedule for sampling is monotonic, i.e., $0 < \hat{\beta}_1 < \ldots < \hat{\beta}_N < 1$, then, for $1 \leq n < N$, $\hat{\beta}_n$ satisfies the following inequality:*

$$0 < \hat{\beta}_n < \min\left\{1 - \frac{\hat{\alpha}^2_{n+1}}{1 - \hat{\beta}_{n+1}}, \hat{\beta}_{n+1}\right\}. \tag{9}$$

The derivation is presented in the Appendix. Note that monotonic noise schedules have been widely used in the prior arts (Song & Ermon, 2019; 2020; Song et al., 2021b) and would lead to a tighter upper bound for $\hat{\beta}_n$. In practice, a tighter bound is helpful to keep the noise schedule short and effective for sampling. Consequently, we define the scheduling network $\sigma_\phi : \mathbb{R}^D \mapsto (0, 1)$ such that

$$\hat{\beta}_n = \min\left\{1 - \frac{\hat{\alpha}^2_{n+1}}{1 - \hat{\beta}_{n+1}}, \hat{\beta}_{n+1}\right\} \sigma_\phi(\boldsymbol{x}_n), \tag{10}$$

where ϕ is a set of learnable neural network parameters. Given the definition of scheduling network, we are now able to estimate the noise schedule for sampling, which we call the *noise scheduling* process. By defining the maximum number of reverse steps (denoted as M), we can sequentially compute the score $\boldsymbol{\epsilon}_\theta\left(\boldsymbol{x}_n, \hat{\alpha}_n\right)$ as in sampling except that the noise scale $\hat{\alpha}_n = \frac{\hat{\alpha}_N}{\prod_{i=n+1}^{N}\sqrt{1-\hat{\beta}_i}}$ is backward computed based on the output of scheduling network starting from the white noise $\boldsymbol{x}_N \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$. The remaining problem is how to learn the network parameter ϕ. We note that $\mathcal{L}_{\text{score}}^{(t)}$ depends on β_t but not $\hat{\beta}_n$, thus it cannot simply be used to learn ϕ. As a result, we first build a link between sampling and training schedules by adding a constraint to the scheduling network:

$$q_\phi(\boldsymbol{x}_{n+1}|\boldsymbol{x}_n = \tilde{\boldsymbol{x}}) = q_{\boldsymbol{\beta}}(\boldsymbol{x}_{t+\tau}|\boldsymbol{x}_t = \tilde{\boldsymbol{x}}), \tag{11}$$

where $1 \leq \tau < T$ is an positive integer, and $\tilde{\boldsymbol{x}}$ is an arbitrary diffused variable. This constraint states that one step of diffusion using $\hat{\beta}_n$ estimated by σ_ϕ equals to τ steps of diffusion using $\beta_t, ..., \beta_{t+\tau}$. Considering the simplest case where $\tau = 1$, i.e., $q_\phi(\boldsymbol{x}_n|\boldsymbol{x}_{n-1}) = q_{\boldsymbol{\beta}}(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$, we can deduce the following proposition for learning ϕ:

Proposition 3. *Assuming θ has been optimized and hypothetically converged to the optimal parameters θ^*, where by optimal parameters it means that $p_{\theta^*}(\boldsymbol{x}_{1:t-1}|\boldsymbol{x}_0) = q_{\boldsymbol{\beta}}(\boldsymbol{x}_{1:t-1}|\boldsymbol{x}_0)$. Then, we can express the gap between $\log p_\theta(\boldsymbol{x}_0)$ and $\mathcal{F}_{score}^{(t)}(\theta^*)$ by ϕ as follows:*

$$\log p_{\theta^*}(\boldsymbol{x}_0) - \mathcal{F}_{score}^{(t)}(\theta^*) = \mathbb{E}_{q_{\boldsymbol{\beta}}(\boldsymbol{x}_t|\boldsymbol{x}_0)}\left[\sum_{i=2}^{t}\mathcal{L}_{step}^{(i)}(\phi;\theta^*)\right], \tag{12}$$

where

$$\mathcal{L}_{step}^{(i)}(\phi;\theta^*) := \mathbb{KL}\left(p_{\theta^*}(\boldsymbol{x}_{i-1}|\boldsymbol{x}_i)||q_\phi(\boldsymbol{x}_{i-1}|\boldsymbol{x}_0)\right). \tag{13}$$

The proof is shown in Appendix. From this proposition, we can observe that minimizing ϕ over $\mathcal{L}_{\text{step}}^{(i)}(\phi;\theta^*)$ leads to minimizing the gap between $\log p_{\theta^*}(\boldsymbol{x}_0)$ and $\mathcal{F}_{\text{score}}^{(t)}(\theta^*)$ even when θ is at its optimal. Besides its practical values, the above proposition is also theoretically appealing. Followed by this proposition, we design a new lower bound $\mathcal{F}_{\text{bddm}}^{(t)}(\theta,\phi)$ that enjoys nice theoretical properties:

Theorem 1. *Relative to the standard ELBO evaluated at step $t \in \{2, ..., T\}$ defined in (Ho et al., 2020) as*

$$\mathcal{F}_{elbo}^{(t)}(\theta) := -\mathbb{E}_{q_{\boldsymbol{\beta}}(\boldsymbol{x}_t|\boldsymbol{x}_0)}\left[\mathbb{KL}\left(q_{\boldsymbol{\beta}}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t,\boldsymbol{x}_0)||p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)\right) + \mathcal{R}_\theta(\boldsymbol{x}_0,\boldsymbol{x}_t)\right], \tag{14}$$

we propose a new lower bound as the following

$$\mathcal{F}_{bddm}^{(t)}(\theta,\phi) := \begin{cases} \mathcal{F}_{score}^{(t)}(\theta) & \text{if} \quad \theta \neq \theta^* \\ \mathcal{F}_{score}^{(t)}(\theta) + \mathbb{E}_{q_{\boldsymbol{\beta}}(\boldsymbol{x}_t|\boldsymbol{x}_0)}\left[\mathcal{L}_{step}^{(t)}(\phi;\theta)\right] & \text{if} \quad \theta = \theta^* \end{cases} \tag{15}$$

which leads to a tighter lower bound when the conditions in Proposition 2 and 3 are satisfied:

$$\log p_\theta(\boldsymbol{x}_0) \geq \mathcal{F}_{bddm}^{(t)}(\theta,\phi) \geq \mathcal{F}_{elbo}^{(t)}(\theta). \tag{16}$$

The proof can be found in Appendix. Note that using $\mathcal{F}_{\text{elbo}}^{(t)}(\theta)$ to train the score network by randomly sampling the diffusion step t at each training step has been proven an effective and efficient training method for score-based DPMs, such as DDPM (Ho et al., 2020), DDIM (Song et al., 2021a) and Improved DDPM (Nichol & Dhariwal, 2021). Theorem 1 suggests that using $\mathcal{F}_{\text{bddm}}^{(t)}(\theta^*,\phi)$ could be a better training method than $\mathcal{F}_{\text{elbo}}^{(t)}(\theta^*)$ towards maximizing the log evidence by additionally learning ϕ with $\mathcal{L}_{\text{step}}^{(t)}(\phi;\theta^*)$. Based on these propositions, we can devise the training and inference algorithms for BDDMs, which are discussed in the following section from a practical point of view.

4 TRAINING, NOISE SCHEDULING, AND INFERENCE

In BDDMs, we train two neural networks: i) a score network $\boldsymbol{\epsilon}_\theta$ for sampling, and ii) a scheduling network σ_ϕ for estimating a noise schedule for sampling. In this section, we first present the training objectives for learning θ and ϕ, respectively, based on the derived propositions.

Algorithm 1 Training Score Network (θ)

Given $T, \{\beta_t\}_{t=1}^T$
Compute $\alpha_t = \prod_{i=1}^t \sqrt{1-\beta_t}$ for $t = 1$ to T
repeat
  $\boldsymbol{x}_0 \sim p_{\text{data}}(\boldsymbol{x}_0)$
  $t \sim \text{Uniform}(\{1, \ldots, T\})$
  $\boldsymbol{\epsilon}_t \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$
  $\boldsymbol{x}_t = \hat{\alpha}_t \boldsymbol{x}_0 + \sqrt{1-\hat{\alpha}_t^2}\boldsymbol{\epsilon}_t$
  $\mathcal{L}_{\text{ddpm}}^{(t)} = \|\boldsymbol{\epsilon}_t - \boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, \alpha_t)\|_2^2$
  Take a gradient descent step on $\nabla_\theta \mathcal{L}_{\text{ddpm}}^{(t)}$
until converged

Algorithm 2 Training Scheduling Network (ϕ)

Given $\theta^*, \tau, T, \{\alpha_t, \beta_t\}_{t=1}^T$
repeat
  $\boldsymbol{x}_0 \sim p_{\text{data}}(\boldsymbol{x}_0)$
  $t \sim \text{Uniform}(\{2, \ldots, T-\tau\})$
  $\hat{\alpha}_n = \alpha_t$, $\hat{\beta}_{n+1} = 1 - (\alpha_{t+\tau}/\alpha_t)^2$
  $\delta_n = \sqrt{1-\hat{\alpha}_n^2}$
  $\boldsymbol{\epsilon}_n \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$
  $\boldsymbol{x}_n = \hat{\alpha}_n \boldsymbol{x}_0 + \delta_n \boldsymbol{\epsilon}_n$
  $\boldsymbol{\epsilon}_{\theta^*}^{(n)} = \boldsymbol{\epsilon}_{\theta^*}(\boldsymbol{x}_n, \hat{\alpha}_n)$
  $\hat{\beta}_n = \min\{\delta_n^2, \hat{\beta}_{n+1}\}\sigma_\phi(\boldsymbol{x}_n)$
  $C_n = 4^{-1}\log(\delta_n^2/\hat{\beta}_n) + 2^{-1}D\left(\hat{\beta}_n/\delta_n^2 - 1\right)$
  $\mathcal{L}_{\text{step}}^{(n)} = \frac{1}{2(\delta_n^2 - \hat{\beta}_n)}\left\|\delta_n\boldsymbol{\epsilon}_n - \frac{\hat{\beta}_n}{\delta_n}\boldsymbol{\epsilon}_{\theta^*}^{(n)}\right\|_2^2 + C_n$
  Take a gradient descent step on $\nabla_\phi \mathcal{L}_{\text{step}}^{(n)}$
until converged

Algorithm 3 Noise Scheduling

Given $\hat{\alpha}_N, \hat{\beta}_N$
$\boldsymbol{x}_N \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$
for $n = N, \ldots, 2$ **do**
  $\boldsymbol{x}_{n-1} \sim p_\theta(\boldsymbol{x}_{n-1}|\boldsymbol{x}_n; \hat{\alpha}_n, \hat{\beta}_n)$
  $\hat{\alpha}_{n-1} = \frac{\hat{\alpha}_n}{\sqrt{1-\hat{\beta}_n}}$
  $\hat{\beta}_{n-1} = \min\{1 - \hat{\alpha}_{n-1}^2, \hat{\beta}_n\}\sigma_\phi(\boldsymbol{x}_{n-1})$
  if $\hat{\beta}_{n-1} < \beta_1$ **then**
    return $\hat{\beta}_n, \ldots, \hat{\beta}_N$
  end if
end for
return $\hat{\beta}_1, \ldots, \hat{\beta}_N$

Algorithm 4 Sampling

Given $\{\hat{\beta}_n\}_{n=1}^N, \boldsymbol{x}_N \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$
Compute $\alpha_n = \prod_{i=1}^n \sqrt{1-\hat{\beta}_n}$ for $n = 1$ to N
for $n = N, \ldots, 1$ **do**
  $\boldsymbol{x}_{n-1} \sim p_\theta(\boldsymbol{x}_{n-1}|\boldsymbol{x}_n; \alpha_n, \hat{\beta}_n)$
end for
return $\boldsymbol{x}_0$

4.1 Training objectives

By Theorem 1, we know that $\mathcal{F}_{\text{bddm}}^{(t)}$ contains two terms: $\mathcal{F}_{\text{score}}^{(t)}(\theta)$ and $\mathcal{L}_{\text{step}}^{(t)}(\phi;\theta)$. On one hand, by Proposition 2, optimizing $\mathcal{F}_{\text{score}}^{(t)}(\theta)$ w.r.t θ can be simplified into minimizing the loss $\mathcal{L}_{\text{ddpm}}^{(t)}(\theta)$ for all $t \in \{1, ..., T\}$, which has been successfully employed in the prior works (Ho et al., 2020). On the other hand, by Proposition 3, we know that ϕ should be trained after θ is well-optimized. In a realistic setting, we can assume the optimized θ^* to be closed to the hypothetically optimal values. By fixing the well-optimized parameters θ^*, we then optimize the proposed loss $\mathcal{L}_{\text{step}}^{(t)}(\phi;\theta^*)$ w.r.t ϕ, which can be simplified into weighted ℓ_2 norms:

$$\mathcal{L}_{\text{step}}^{(t)}(\phi;\theta^*) = \frac{1}{2(1-\hat{\beta}_t(\phi)-\alpha_t^2)}\left\|\sqrt{1-\alpha_t^2}\boldsymbol{\epsilon}_t - \frac{\hat{\beta}_t(\phi)}{\sqrt{1-\alpha_t^2}}\boldsymbol{\epsilon}_{\theta^*}(\boldsymbol{x}_t, \alpha_t)\right\|_2^2 + C_t(\phi), \tag{17}$$

where

$$C_t(\phi) := \frac{1}{4}\log\frac{1-\alpha_t^2}{\hat{\beta}_t(\phi)} + \frac{D}{2}\left(\frac{\hat{\beta}_t(\phi)}{1-\alpha_t^2} - 1\right). \tag{18}$$

The simplification procedures can be found in Appendix.

4.2 Efficient training

Suggested by Proposition 3, we know that θ should be optimized before learning ϕ. For training the score network $\boldsymbol{\epsilon}_\theta$, similar to the prior settings in (Ho et al., 2020; Chen et al., 2020; Song et al., 2021a), we define $\boldsymbol{\beta}$ as a linear noise schedule:

$$\beta_t = \frac{\varepsilon}{T-t+1}, \quad \text{for} \quad 1 \le t \le T, \tag{19}$$

where ε is a hyperparameter that specifies the maximum value in this linear schedule, i.e., β_T. By Proposition 2, we can sample $t \sim \text{Uniform}(\{1, ..., T\})$ for evaluating $\mathcal{L}_{\text{ddpm}}^{(t)}$, resulting in the same Algorithm 1 as in (Ho et al., 2020).

Next, given a well-optimized θ, we can start training the scheduling network σ_ϕ. By the definition of σ_ϕ in Eq. (10), for training we need to first obtain the noise scale from the preceding step, i.e., $\hat{\beta}_{n+1}$. To achieve this, we randomly sample the noise scale as follows:

$$\hat{\beta}_{n+1} \sim \text{Uniform}\left(\left\{1 - (\alpha_{t+\tau}/\alpha_t)^2 \,|\, t \in \{2, ..., T-\tau\}\right\}\right), \tag{20}$$

where $0 < \tau < T$ is a hyperparameter defined in Eq. (11) to control the inference step size relative to the noise schedule for training $\boldsymbol{\beta}$. Expectedly, a larger τ should lead to a shorter noise schedule $\hat{\boldsymbol{\beta}}$. Here, we note that Theorem 1 still holds even if we change the index from t to n. Followed from this sampling strategy, we can efficiently train the scheduling network, as presented in Algorithm 2. Our empirical observation in section 5 shows that, although a linear schedule is used to define $\boldsymbol{\beta}$ to sample the local transition from $n+1$ to n, the predicted schedule $\hat{\boldsymbol{\beta}}$ is not limited into a linear one.

4.3 Noise scheduling for fast and high-quality sampling

After the score network and the scheduling network are trained, BDDMs split the inference procedure into two phases: (1) the noise scheduling phase and (2) the sampling phase. First, we consider sampling from a reverse process with N iterations maximum to obtain a noise schedule for sampling $\hat{\boldsymbol{\beta}}$. Different from the forward computed α_t which is fixed to start from $\sqrt{1-\beta_1}$, $\hat{\alpha}_n$ is a backward computed variable (from N to 1) that may deviates from the forward one because $\{\hat{\beta}_i\}_i^{n-1}$ are unknown in the noise scheduling phase. Therefore, for noise scheduling, we have two hyperparameters – $(\hat{\alpha}_N, \hat{\beta}_N)$. Overall, the noise scheduling procedure is presented in Algorithm 3. Note that we use β_1, the smallest noise scale seen in training, as a threshold to determine when to stop the noise scheduling process, so that we can ignore the small noise scales ($< \beta_1$) unseen by the score network that may lead to a numerical issue. By doing so, we can also get a reasonably shorter $\hat{\boldsymbol{\beta}}$.

In practice, we apply a grid search algorithm of M bins to find a good set of initial values for $(\hat{\alpha}_N, \hat{\beta}_N)$, which takes $\mathcal{O}(M^2)$ time. This is computationally feasible in contrast to the unscalable $\mathcal{O}(M^N)$ grid search algorithm used in (Chen et al., 2020) that searches over the entire noise schedule $\hat{\boldsymbol{\beta}}$. Empirically, the grid search for our noise scheduling algorithm can be evaluated on a small subset of the training samples (as few as 1 sample) to find a generally well-performed noise schedule.

After the noise scheduling procedure is finished, we fix the noise schedule $\hat{\boldsymbol{\beta}} \in \mathbb{R}^N$ to the best search result for the subsequent sampling over the whole testing set. The sampling procedure is the same as in (Ho et al., 2020). By initializing the white noise $\boldsymbol{x}_N \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$, we can generate $\boldsymbol{x}_0$ after N iterations of the reverse process, as shown in Algorithm 4.

5 Experiments

We conducted a series of experiments on neural vocoding to evaluate the proposed BDDMs. First, we compared BDDMs against the best publicly available models: the mixture of logistics (MoL) WaveNet (Oord et al., 2018) implemented in (Yamamoto, 2020), the WaveGlow (Prenger et al., 2019) implemented in (Valle, 2020), the MelGAN (Kumar et al., 2019a) implemented in (Kumar, 2019), the HiFi-GAN (Kong et al., 2020b) implemented in (Kong et al., 2020a) and the two most recently proposed diffusion-based vocoders, i.e., WaveGrad (Chen et al., 2020) and DiffWave (Kong et al., 2021), which were both re-implemented in our code.

In addition, we also compared BDDMs to a variety of scheduling and acceleration techniques applicable to DDPMs, including the grid search (GS) approach in WaveGrad, the fast sampling (FS) approach based on a user-defined 6-step schedule in DiffWave, the denoising diffusion implicit model (DDIM) (Song et al., 2021a) and a noise estimation (NE) approach (San-Roman et al., 2021). We detail our implementation of each of these methods in the Appendix.

For fair and reproducible comparison with other models and approaches, we used the LJSpeech dataset (Ito & Johnson, 2017), which consists of 13,100 22kHz audio clips of a female speaker. All diffusion models were trained on the same LJ speech training set as used in (Chen et al., 2020). The hyperparameter setting of BDDMs is presented in the Appendix.

Table 1: Comparison of neural vocoders in terms of MOS with 95% confidence intervals, real-time factor (RTF) and model size in megabytes. Each MOS was undertaken a student-t test to see if it is significantly different from the score of ground truth. We use bold figures for p-values ≥ 0.05.

Neural Vocoder	**MOS** (p-value)	**RTF**	**Size**
Ground-truth	4.62 ± 0.24	—	—
WaveNet (MoL) (Oord et al., 2018)	3.63 ± 0.26 (1.05×10^{-9})	318.6	282MB
WaveGlow (Prenger et al., 2019)	3.06 ± 0.37 (1.10×10^{-10})	0.0198	645MB
MelGAN (Kumar et al., 2019a)	3.52 ± 0.35 (1.08×10^{-8})	0.00396	17MB
HiFi-GAN (Kong et al., 2020b)	4.32 ± 0.25 (3.25×10^{-3})	0.0134	54MB
WaveGrad - 1000 steps (Chen et al., 2020)	4.39 ± 0.23 (0.012)	38.2	183MB
DiffWave - 200 steps (Kong et al., 2021)	$\mathbf{4.52 \pm 0.24}$ (0.161)	7.30	27MB
BDDM - 3 steps ($\hat{\alpha}_N = 0.68$, $\hat{\beta}_N = 0.53$)	3.65 ± 0.23 (4.58×10^{-10})	0.110	27MB
BDDM - 7 steps ($\hat{\alpha}_N = 0.62$, $\hat{\beta}_N = 0.42$)	$\mathbf{4.43 \pm 0.31}$ (0.055)	0.256	27MB
BDDM - 12 steps ($\hat{\alpha}_N = 0.67$, $\hat{\beta}_N = 0.12$)	$\mathbf{4.47 \pm 0.33}$ (0.102)	0.438	27MB

Table 2: Comparison of sampling acceleration methods with the same score network and the same number of steps. Each score was undertaken a student-t test to see if it is significantly higher than the others within the same number of steps. We use bold figures for p-values < 0.05.

Steps	**Acceleration Method**	**STOI**	**PESQ**	**MOS**
3	GS (Chen et al., 2020)	$\mathbf{0.965 \pm 0.009}$	$\mathbf{3.66 \pm 0.20}$	
	FS (Kong et al., 2021)	0.939 ± 0.023	3.09 ± 0.23	
	DDIM (Song et al., 2021a)	0.943 ± 0.015	3.42 ± 0.27	
	NE (San-Roman et al., 2021)	$\mathbf{0.966 \pm 0.010}$	$\mathbf{3.62 \pm 0.18}$	
	BDDM	$\mathbf{0.966 \pm 0.011}$	$\mathbf{3.63 \pm 0.24}$	3.65 ± 0.22
7	FS (Kong et al., 2021)	$\mathbf{0.981 \pm 0.006}$	3.68 ± 0.24	
	DDIM (Song et al., 2021a)	0.974 ± 0.008	3.85 ± 0.12	
	NE (San-Roman et al., 2021)	0.978 ± 0.007	3.75 ± 0.18	
	BDDM	$\mathbf{0.983 \pm 0.006}$	$\mathbf{3.96 \pm 0.09}$	$\mathbf{4.43 \pm 0.30}$
12	DDIM (Song et al., 2021a)	0.979 ± 0.006	3.90 ± 0.10	
	NE (San-Roman et al., 2021)	0.981 ± 0.007	3.82 ± 0.13	
	BDDM	$\mathbf{0.987 \pm 0.006}$	$\mathbf{3.98 \pm 0.12}$	$\mathbf{4.47 \pm 0.31}$

5.1 Sampling quality in objective and subjective metrics

Since the objective and subjective metrics have been widely applied and well-established for speech quality evaluation, we used both measures to evaluate the generative models on speech vocoding. Specifically, we used two scale-invariant metrics: the perceptual evaluation of speech quality (PESQ) (Rix et al., 2001) and the short-time objective intelligibility (STOI) (Taal et al., 2010) to measure the noisiness and the distortion of the generated speech relative to the reference speech. Mean opinion score (MOS) was used as the subjective metric to assess the naturalness of the speech generated by different neural vocoders given the same ground truth spectrograms. The assessment scheme of MOS is included in the Appendix.

Firstly, we compared BDDMs against the state-of-the-art (SOTA) vocoders in Table 1, in which three hyperparameter sets searched for BDDMs corresponding to 3, 7 and 12 sampling steps are listed. Out of the 9 vocoders, only DiffWave with 200 steps and our proposed BDDMs with 7 and 12 steps showed no statistically significant difference from the ground-truth in terms of MOS. Yet, BDDMs outperformed DiffWave in terms of RTFs. Notably, while previous diffusion-based vocoders achieved the highest MOS scores, they struggled to meet an acceptable RTF for industrial deployment. In contrast, BDDM with only 7 sampling steps (143x faster than WaveGrad and 28.6x faster than DiffWave) still can remain a high standard of generation performance.

Secondly, we compared BDDMs against alternative accelerated sampling methods as shown in Table 2, where we fixed the number of sampling steps searched by BDDMs, i.e., 3, 7 and 12 steps, and

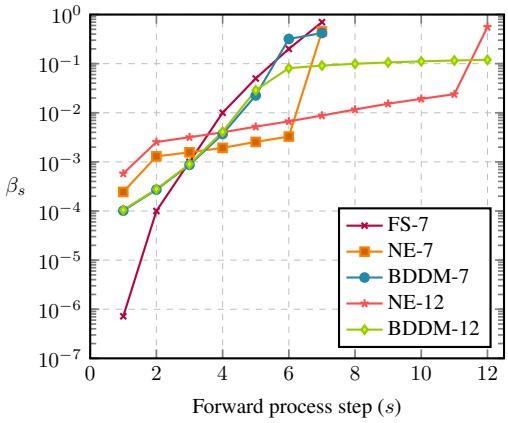

Figure 1: Noise schedules in log-scale y axis

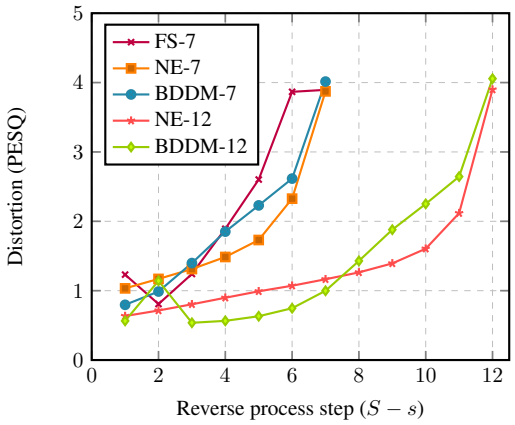

Figure 2: PESQ scores of different methods

used the same score network for a pair-to-pair comparison. Overall, the GS method performed stably with the step number is small (i.e., $N \leq 6$ in practice) but is not scalable; the FS method was initially defined as a fixed 6-step noise schedule, which was linearly interpolated to 3 steps and 7 steps, respectively, yet failed to remain stable performances; both the DDIM and the NE methods were stable across all steps but are not competitive enough. In comparison, BDDMs consistently attained the leading scores in all three step numbers. From which, we confirm that BDDM is superior than other acceleration methods for DPMs in terms of both stability and quality.

5.2 Analyzing noise scheduling behaviors

For better understanding the noise scheduling behavior of BDDMs, we compared the estimated noise schedules generated by different methods in a log-scale plot, as illustrated in Figure 1, in which the number after the method name is referred to as steps of the model as presented in Table 2. We ask the readers to relate the performance of each model in Table 2 to the plot of its actual noise schedule. We can observe that the "NE-12" had the "flattest" schedule, which may explain why it also had the lowest scores in both objective and subjective metrics. In contrast, the best performing "BDDM-12" estimated the steepest schedule and, interestingly, devised a turning point after the noise scale went greater than about 0.01. We also observed a less obvious turning behavior from the noise schedules of "BDDM-7" after passing 0.1. Before the turning point, our BDDMs tended to keep the noise scales as a geometric sequence. We conceive that such behavior is impressive as the step-index was unknown to the scheduling network σ_ϕ. In Figure 2, we also compared the change of PESQ as a reliable measure of distortion, as the sampling index s started from S to 1 on the same held-out example among different methods. Besides having the highest final PESQ score, BDDMs quickly surpassed all other approaches after about a half of the total sampling steps.

5.3 Ablation studies

We attribute the primary superiority of BDDMs to the derived loss $\mathcal{L}_{\text{step}}^{(t)}$ for learning ϕ. We performed an ablation study in Appendix to use the standard negative ELBO $-\mathcal{F}_{\text{elbo}}^{(t)}$ in place of $\mathcal{L}_{\text{step}}^{(t)}$ for learning ϕ. However, it turned out that the scheduling network trained with $-\mathcal{F}_{\text{elbo}}^{(t)}$ rapidly collapsed to output zero within several training steps, whereas the scheduling network trained with $\mathcal{L}_{\text{step}}^{(t)}$ reasonably outputted fluctuating values around one.

6 Conclusions

In this paper, we introduced the bilateral denoising diffusion models (BDDMs), a novel generative model that simultaneously parameterized the forward process and the reverse process by two specific networks: the score network $\boldsymbol{\epsilon}_\theta$ and the scheduling network σ_ϕ, respectively. We derived a new

lower bound to the log evidence for learning θ that can be efficiently computed for one time step t rather than processing though T time steps. We also showed that the new objective could lead to the same loss derived in DDPMs (Ho et al., 2020) under a reasonable condition. We then derived another training objective for learning ϕ, which can be viewed as tightening the lower bound when θ is optimized. Based on the new objectives, we designed a new lower bound tighter than the standard evidence lower bound. Moreover, an efficient training algorithm and a noise scheduling algorithm were respectively presented based on the propositions. Finally, we demonstrated the superiority of BDDMs over previous denoising diffusion models in terms of both generation quality and sampling speed.

REFERENCES

Mikołaj Bińkowski, Jeff Donahue, Sander Dieleman, Aidan Clark, Erich Elsen, Norman Casagrande, Luis C. Cobo, and Karen Simonyan. High fidelity speech synthesis with adversarial networks. *ICLR*, 2020.

Sam Bond-Taylor, Adam Leach, Yang Long, and Chris G. Willcocks. Deep generative modelling: A comparative review of vaes, gans, normalizing flows, energy-based and autoregressive models. *https://arxiv.org/abs/2103.04922*, 2021.

Miguel A Carreira-Perpinan and Geoffrey E Hinton. On contrastive divergence learning. *AISTATS*, pp. 33–40, 2005.

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.

Tian Qi Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *NeurIPs*, pp. 6571–6583, 2018.

Prafulla Dhariwal and Alex Nichol. Diffusion models beat gans on image synthesis. *arXiv preprint arXiv:2105.05233*, 2021.

Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*, 2016.

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *arXiv:1406.2661*, 2014.

G. E. Hinton. Training products of experts by minimizing contrastive divergence. neural computation. *Neural computation*, pp. 14(8):1771–1800, 2002.

Jonathan Ho, Xi Chen, Aravind Srinivas, Yan Duan, and Pieter Abbeel. Flow++: Improving flow-based generative models with variational dequantization and architecture design. *ICML*, 2019.

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

Aapo Hyvarinen and Peter Dayan. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, pp. 6(4), 2005.

Keith Ito and Linda Johnson. The lj speech dataset. *https://keithito.com/LJ-Speech-Dataset/*, 2017.

Nal Kalchbrenner, Erich Elsen, Karen Simonyan, Seb Noury, Norman Casagrande, Edward Lockhart, Florian Stimberg, Aaron Oord, Sander Dieleman, and Koray Kavukcuoglu. Efficient neural audio synthesis. In *International Conference on Machine Learning*, pp. 2410–2419. PMLR, 2018.

Diederik P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. *NeurIPS*, pp. 10215–10224, 2018.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Jungil Kong, Jaehyeon Kim, and Jaekyoung Bae. Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis. https://github.com/jik876/hifi-gan, 2020a.

Jungil Kong, Jaehyeon Kim, and Jaekyoung Bae. Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis. *Advances in Neural Information Processing Systems*, 33, 2020b.

Zhifeng Kong and Wei Ping. On fast sampling of diffusion probabilistic models. *arXiv preprint arXiv:2106.00132*, 2021.

Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. Diffwave: A versatile diffusion model for audio synthesis. *ICLR*, 2021.

Kundan Kumar, Rithesh Kumar, Thibault de Boissiere, Lucas Gestin, Wei Zhen Teoh, Jose Sotelo, Alexandre de Brébisson, Yoshua Bengio, and Aaron Courville. Melgan: Generative adversarial networks for conditional waveform synthesis. *arXiv preprint arXiv:1910.06711*, 2019a.

Kundan Kumar, Rithesh Kumar, Thibault de Boissiere, Lucas Gestin, Wei Zhen Teoh, Jose Sotelo, Alexandre de Brebisson, Yoshua Bengio, and Aaron Courville. Melgan: Generative adversarial networks for conditional waveform synthesis. *NeurIPS*, 2019b.

Rithesh Kumar. Official repository for the paper melgan: Generative adversarial networks for conditional waveform synthesis. https://github.com/descriptinc/melgan-neurips, 2019.

Eric Luhman and Troy Luhman. Knowledge distillation in iterative generative models for improved sampling speed. *arXiv:2101.02388*, 2021.

Lars Maaløe, Marco Fraccaro, Valentin Liévin, and Ole Winther. Biva: A very deep hierarchy of latent variables for generative modeling. *NeurIPS*, pp. 6548–6558, 2019.

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

Aaron Oord, Yazhe Li, Igor Babuschkin, Karen Simonyan, Oriol Vinyals, Koray Kavukcuoglu, George Driessche, Edward Lockhart, Luis Cobo, Florian Stimberg, et al. Parallel wavenet: Fast high-fidelity speech synthesis. In *International conference on machine learning*, pp. 3918–3926. PMLR, 2018.

George Papamakarios, Eric Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. Normalizing flows for probabilistic modeling and inference. *JMLR*, pp. 22(57):1–64, 2021.

Ryan Prenger, Rafael Valle, and Bryan Catanzaro. Waveglow: A flow-based generative network for speech synthesis. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3617–3621. IEEE, 2019.

Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. *ICML*, pp. 1278–1286, 2014.

Antony W Rix, John G Beerends, Michael P Hollier, and Andries P Hekstra. Perceptual evaluation of speech quality (pesq)-a new method for speech quality assessment of telephone networks and codecs. In *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No. 01CH37221)*, volume 2, pp. 749–752. IEEE, 2001.

Robin San-Roman, Eliya Nachmani, and Lior Wolf. Noise estimation for generative diffusion models. *arXiv preprint arXiv:2104.02600*, 2021.

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

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *ICLR*, 2021a.

Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *NeurIPS*, 2019.

Yang Song and Stefano Ermon. Improved techniques for training score-based generative models. *arXiv preprint arXiv:2006.09011*, 2020.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *ICLR*, 2021b.

Cees H Taal, Richard C Hendriks, Richard Heusdens, and Jesper Jensen. A short-time objective intelligibility measure for time-frequency weighted noisy speech. In *2010 IEEE international conference on acoustics, speech and signal processing*, pp. 4214–4217. IEEE, 2010.

Rafael Valle. Waveglow: a flow-based generative network for speech synthesis. `https://github.com/NVIDIA/waveglow`, 2020.

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. *Proc. 9th ISCA Speech Synthesis Workshop*, pp. 125–125, 2016.

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

Ryuichi Yamamoto. Wavenet vocoder. `https://github.com/r9y9/wavenet_vocoder`, 2020.

Ryuichi Yamamoto, Eunwoo Song, and Jae-Min Kim. Parallel. Wavegan: A fast waveform generation model based on generative adversarial networks with multi-resolution spectrogram. *ICASSP*, 2020.

Geng Yang, Shan Yang, Kai Liu, Peng Fang, Wei Chen, and Lei Xie. Multi-band melgan: Faster waveform generation for high-quality text-to-speech. *arXiv preprint arXiv:2005.0510*, 2020.