Cauchy Diffusion: A Heavy-tailed Denoising Diffusion Probabilistic Model for Speech Synthesis

Qi Lian[1], **Yu Qi**[2,3,1,4*], **Yueming Wang**[1]

[1] The College of Computer Science and Technology, Zhejiang University, China
[2] MOE Frontier Science Center for Brain Science and Brain-machine Integration, Zhejiang University, China
[3] Affiliated Mental Health Center & Hangzhou Seventh People's Hospital, Zhejiang University, China
[4] State Key Lab of Brain-Machine Intelligence, Zhejiang University, China
{lianqi, qiyu, ymingwang}@zju.edu.cn

Abstract

Denoising diffusion probabilistic models (DDPMs) have gained popularity in devising neural vocoders and obtained outstanding performance. However, existing DDPM-based neural vocoders struggle to handle the prosody diversities due to their susceptibility to mode-collapse issues confronted with imbalanced data. We introduced Cauchy Diffusion, a model incorporating the Cauchy noises to address this challenge. The heavy-tailed Cauchy distribution exhibits better resilience to imbalanced speech data, potentially improving prosody modeling. Our experiments on the LJSpeech and VCTK datasets demonstrate that Cauchy Diffusion achieved state-of-the-art speech synthesis performance. Compared to existing neural vocoders, our Cauchy Diffusion notably improved speech diversity while maintaining superior speech quality. Remarkably, Cauchy Diffusion surpassed neural vocoders based on generative adversarial networks (GANs) that are explicitly optimized to improve diversity.

Introduction

Speech synthesis is a comprehensive technique that enables machines to produce authentic speech, drawing from various disciplines within natural language processing, such as acoustics and linguistics (Ning et al. 2019). Deep generative models have recently made significant advancements in speech synthesis, encompassing flow-based models (Prenger, Valle, and Catanzaro 2019), GAN-based models (Kong, Kim, and Bae 2020; Jang et al. 2021), and DDPMs (Chen et al. 2020; Kong et al. 2020; Huang et al. 2022). Current generative models for speech synthesis typically comprise two components: an encoder that extracts acoustic or linguistic features from the input text and a neural vocoder that converts these features into raw waveforms (Wang et al. 2017). DDPMs have attracted notable attention among these models for their exceptional generative capabilities in devising neural vocoders (Yang et al. 2023; Cao et al. 2024).

However, DDPMs are susceptible to mode-collapse issuses when dealing with imbalanced data (Qin et al. 2023). The diversity of prosodies arising from various physical conditions and environments make the same spoken sentence exhibit completely different prosodies, which leads to an imbalance in speech data (Nooteboom et al. 1997). The positions of silent pauses and pitch variations could be unclear, and the waveforms of identical words or sentences may vary greatly (McPherson and McDermott 2018). Existing DDPM-based neural vocoders struggle to handle the diversity of prosodies.

To address these issues, we explore the viability of employing the Cauchy distribution as a substitute for DDPM-based neural vocoders. Unlike the Gaussian distribution, the Cauchy distribution has heavier tails, making it more tolerant of imbalanced distributions (Feller 1991; Foss et al. 2011; Qi et al. 2014). Previous studies have revealed that utilizing a heavy-tailed distribution in deep generative models helps improve the diversity of synthesized samples (YOON et al. 2023; Han et al. 2024). By incorporating Cauchy noise into the DDPM, we anticipate potential improvements in the prosody diversities of synthesized speeches.

Designing a DDPM with Cauchy distribution requires the implementation of diffusion and generative processes using Cauchy noises. Furthermore, since stochastic sampling provides better diversity than deterministic sampling, we aim to devise a DDPM that supports stochastic sampling by utilizing noise and its scale at each diffusion step to maximize prosody diversities. The main challenge is that the Cauchy distribution does not adhere to the Kolmogorov equations.

Inspired by the definition of ratio distribution, we represent the Cauchy distribution using two Gaussian distributions (Marsaglia 1965). Two Gaussian distributions and one Cauchy distribution are unified under the same location-scale parameterization. The scales of the Cauchy noises in the diffusion and generative processes can be determined through ratio distribution. An optimization objective function that minimizes the differences between the Cauchy noises and their scales can then be defined. Lastly, we discovered that the sampling methodology introduced in (Song, Meng, and Ermon 2020) performs well for our model, owing to the similarity between the Gaussian and Cauchy distributions. Using this methodology, our model supports deterministic and stochastic sampling processes. The resulting denoising diffusion probabilistic model incorporating Cauchy noises is termed as Cauchy Diffusion.

The experimental results on the LJSpeech and VCTK datasets confirmed the effectiveness of our approach. Our

*corresponding author.

approach yielded state-of-the-art performance of prosody diversity on both datasets. Furthermore, our approach obtained superior performance on both datasets regarding objective quality metrics and subjective listening tests.

The Cauchy Diffusion Model

This section delves into the details of the Cauchy Diffusion model, a DDPM incorporating heavy-tailed Cauchy noises. First, we define the Cauchy diffusion process by unifying the Cauchy distribution and the Gaussian distribution in the same location-scale parameterization. Next, we derive the Cauchy diffusion schedule that determines the prior and posterior scales of the Cauchy noises by formulating the Cauchy distribution as a ratio distribution of two Gaussian distributions. Then, we introduce the training objective that minimizes the discrepancies between the added and predicted Cauchy noises and their corresponding scales. Lastly, we provide details of a viable sampling method for the Cauchy Diffusion model.

Background

The target of the diffusion model is to learn a distribution $p_\theta(\mathbf{x}_0)$ parameterized by θ from samples drawn from an unknown data distribution $q(\mathbf{x}_0)$ (Ho, Jain, and Abbeel 2020; Song, Meng, and Ermon 2020; Nichol and Dhariwal 2021). Denoising diffusion probabilistic models are latent variable models that can be defined as

$$p_\theta(\mathbf{x}_0) := \int p_\theta(\mathbf{x}_{0:T}) d\mathbf{x}_{1:T} \tag{1}$$

where $\mathbf{x}_1, \ldots, \mathbf{x}_T$ are latent variables. The denoising diffusion model consists of the diffusion process and the reverse process. The *diffusion process* is a Markov chain that gradually adds noise to the data according to a predefined schedule $\beta = \{\beta_1, \ldots, \beta_T\}$. Mathematically, the process can be defined as

$$q(\mathbf{x}_{1:T}|\mathbf{x}_0) := \prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1}) \tag{2}$$

In contrast, the *reverse process* gradually removes the added noise based on a schedule derived from the predefined schedule β. The reverse process can be defined as

$$p_\theta(\mathbf{x}_{0:T}) := p(\mathbf{x}_T) \prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) \tag{3}$$

Cauchy Diffusion Process

The Cauchy distribution is a member of the location-scale family, which is defined as a class Ω of probability distributions. In this family, for any cumulative distribution function $F \in \Omega$ and any real number $a \in \mathbb{R}$ and $b > 0$, the distribution function $G(x) = F(a + bx)$ is also a member of Ω. A location-scale distribution is parameterized by a location and a non-negative scale. In the context of the Cauchy diffusion process, let β_t and ϵ_t denote the squared scale (predefined schedule) of a Cauchy distribution and a standard Cauchy noise sampled at diffusion step t, respectively. As shown in Figure 1, since the Cauchy distribution is closed under linear transformation, the conditional distribution $q(\mathbf{x}_t|\mathbf{x}_{t-1})$ of the Cauchy distribution can be defined as

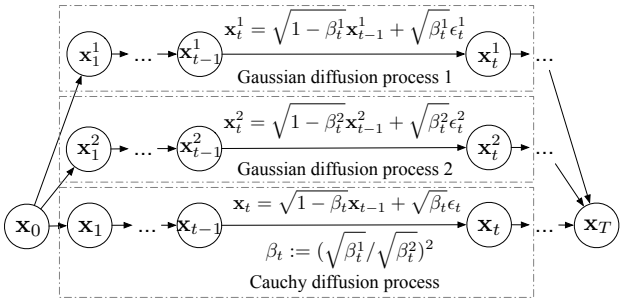

Figure 1: A Cauchy diffusion process can be constructed using two Gaussian diffusion processes by the definition of the ratio distribution.

$$\mathbf{x}_t = \sqrt{1-\beta_t}\mathbf{x}_{t-1} + \sqrt{\beta_t}\epsilon_t \tag{4}$$

The same reparametrization trick in Gaussian DDPMs can be applied to the Cauchy diffusion process. Defining $\alpha_t := 1 - \beta_t$ and $\bar{\alpha}_t := \prod_{s=1}^{t} \alpha_s$, the conditional distribution $q(\mathbf{x}_t|\mathbf{x}_0)$ of the Cauchy diffusion process can be defined as

$$\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon_t \tag{5}$$

Cauchy Diffusion Schedule

The Cauchy Diffusion model should support stochastic sampling to improve the prosody diversities of generated speeches, which requires scale prediction in the generative process. Researchers have recently proposed many diffusion schedules, including the linear schedule (Ho, Jain, and Abbeel 2020), cosine schedule (Nichol and Dhariwal 2021), sigmoid schedule (Jabri, Fleet, and Chen 2022), and so on (Chen 2023). However, these diffusion schedules can not directly apply to our Cauchy Diffusion model. Since the Cauchy distribution does not follow Kolmogorov equations, the posterior scale can not be computed analytically in the diffusion process, and scale prediction becomes impossible in the generative process. This section explains the steps of developing a diffusion schedule for the Cauchy Diffusion model.

Scale Computation Both the Cauchy distribution and the Gaussian distribution belong to the location-scale family, and they can be parameterized in the same way by recognizing the variance of the Gaussian distribution as the squared scale. The Cauchy distribution comes as the ratio of two normally distributed variables with zero mean (Marsaglia 1965). We can use the ratio distribution of two Gaussian diffusion processes that follow Kolmogorov equations to construct the schedule for the Cauchy diffusion process. Let $\beta^1 = \{\beta_1^1, \ldots, \beta_t^1, \ldots, \beta_T^1\}$ represents the squared scales of the first Gaussian DDPM, and $\beta^2 = \{\beta_1^2, \ldots, \beta_t^2, \ldots, \beta_T^2\}$ represents the squared scales of the second Gaussian DDPM. The squared scale of Cauchy diffusion at diffusion step t can be defined as

$$\beta_t := (\sqrt{\beta_t^1}/\sqrt{\beta_t^2})^2 \tag{6}$$

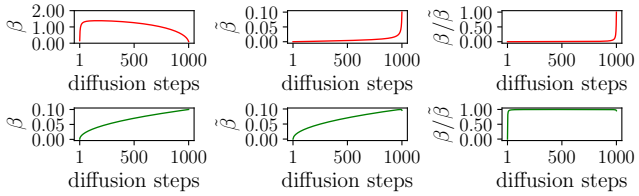

Figure 2: Reformulate the scale computation produces applicable prior and posterior scales for the Cauchy Diffusion model. The first and second rows depict the prior (β) and the posterior scales ($\tilde{\beta}$) and their ratios ($\beta/\tilde{\beta}$) before and after reformulation, respectively.

Defining $\alpha_t^1 := 1 - \beta_t^1$ and $\bar{\alpha}_t^1 := \prod_{i=1}^{t} \alpha_i^1$ for the first Gaussian DDPM, and $\alpha_t^2 := 1 - \beta_t^2$ and $\bar{\alpha}_t^2 := \prod_{i=1}^{t} \alpha_i^2$ for the second Gaussian DDPM. The posterior squared scales for the two Gaussian DDPMs at diffusion step t are given by $\tilde{\beta}_t^1 = (1 - \bar{\alpha}_{t-1}^1)/(1 - \bar{\alpha}_t^1)\beta_t^1$ and $\tilde{\beta}_t^2 = (1 - \bar{\alpha}_{t-1}^2)/(1 - \bar{\alpha}_t^2)\beta_t^2$. The posterior squared scale of Cauchy diffusion at diffusion step t can then be defined as

$$\tilde{\beta}_t := (\sqrt{\tilde{\beta}_t^1}/\sqrt{\tilde{\beta}_t^2})^2 \tag{7}$$

Reformulation of the Scale Computation Empirically, we have found that computing the prior scale of the Cauchy diffusion process by dividing the prior scales of two Gaussian diffusion processes leads to pathological results. As shown in the top row of Figure 2, the prior scale of the Cauchy diffusion process is unexpectedly high, which makes the ratios of the prior and posterior scales of the Cauchy diffusion process approaching zero almost everywhere, except the last few steps. We can reformulate the prior scales ' computation since we are optimizing the denoising model according to the Cauchy diffusion schedule instead of the Gaussian diffusion schedule. We first define the schedules of the Cauchy diffusion and one Gaussian diffusion and then derive the schedule of the second Gaussian diffusion schedule. Specifically, the prior scale of the second Gaussian diffusion process is defined by

$$\sqrt{\beta_t^2} = \sqrt{\beta_t} * \sqrt{\beta_t^1} \tag{8}$$

Given the prior scale, we can analytically compute the second Gaussian diffusion process's posterior scale. Further, we can compute the posterior scales of the Cauchy diffusion process using using Equation(7). As shown in the bottom row of Figure 2, the reformulation produces similar prior and posterior scales of the Cauchy Diffusion model, and their ratios are healthy. Note that, the two known diffusion schedules used to plot Figure 2 are generated by a linear scheme with $\beta_0 = 1e-6$ and $\beta_T = 1e-2$, and a cosine scheme with $s = 8e-3$, respectively.

Training Objective

The parameters θ are tuned to fit the data distribution $q(\mathbf{x}_0)$ by optimizing a variational bound on the negative log-likelihood defined as

$$\mathbb{E}[-\log p_\theta(\mathbf{x}_0)] \leq \mathbb{E}_q[-\log \frac{p_\theta(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}] \tag{9}$$

The variational bound can be decomposed to $L := L_0 + L_1 + \cdots + L_T$, where $L_0 := -\log p_\theta(\mathbf{x}_0|\mathbf{x}_1)$, $L_{t-1} := D_{KL}(q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)||p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t))$, and $L_T := D_{KL}(q(\mathbf{x}_T|\mathbf{x}_0)||p(\mathbf{x}_T))$. As the evaluation of L_0 depends on the task, and L_T is independent of θ, the loss terms $\{L_t\}_{t=1}^{T-1}$ play a crucial role in tuning the parameters θ.

Researchers have proposed many prediction targets for DDPMs, such as the input $\mathbf{x}_0$, the added noise ϵ_t and the corrupted input $\mathbf{x}_t$ at diffusion step t (Ho, Jain, and Abbeel 2020). Since the Cauchy Diffusion model share the same diffusion process as the Gaussian DDPMs, we can predict the added Cauchy noise ϵ_t, and calculate the input $\mathbf{x}_0$ given $\mathbf{x}_t$ using the following equation:

$$\mathbf{x}_0 = \frac{1}{\alpha_t}(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_t) \tag{10}$$

The objective of predicting the added Cauchy noise can be defined as

$$L_\gamma(\epsilon_\theta) := \sum_{t=1}^{T} \gamma_t \mathbb{E}_{\mathbf{x}_0,t}[||\epsilon_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon_t, t) - \epsilon_t||_2^2] \tag{11}$$

where ϵ_t denotes the sampled Cauchy noise and $\epsilon_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon_t, t)$ denotes the predicted Cauchy noise at diffusion step t. By setting all the coefficients γ_t to 1, the objective function can be simplified to

$$L(\epsilon_\theta) := \sum_{t=1}^{T} \mathbb{E}_{\mathbf{x}_0,t}[||\epsilon_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon_t, t) - \epsilon_t||_2^2] \tag{12}$$

The simplified objective fails to provide error signals for the scales, and an improved training objective that simultaneously predicts the added noise and the corresponding squared scales has been proposed to address this problem (Nichol and Dhariwal 2021). In this objective, we can analytically compute the Gaussian distribution's posterior location to estimate the KL-divergence between two Gaussian distributions. However, the Cauchy Diffusion does not follow the Kolmogorov equations, and we can not analytically compute the posterior location. Motivated by the stop-gradient operation applied in (Nichol and Dhariwal 2021), we propose calculating the KL-divergence between two Cauchy distributions with locations equal to zero. This approach is viable because the posterior squared scales of our Cauchy Diffusion model can be estimated using Equation (7). The KL-divergence terms L_t between two Cauchy distributions (Chyzak and Nielsen 2019) can be defined as

$$L_t := \log \frac{(\tilde{\beta}_t + \beta_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon_t, t))^2}{4\tilde{\beta}_t\beta_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon_t, t)} \tag{13}$$

where $\beta_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon_t, t)$ denotes the predicted Cauchy squared scale. We interpolate the squared scale of the Cauchy Diffusion model in the log domain through

$$\beta_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon_t, t) = \exp(sigmoid(v)\log\beta_{t-1} + (1 - sigmoid(v))\log\tilde{\beta}_{t-1}) \tag{14}$$

where v refers to the output of the denoising network. The KL-divergence loss can be defined as

$$L(\beta_\theta) := \sum_{t=1}^{T-1} L_t \tag{15}$$

In summary, the training objective of the Cauchy Diffusion model can be defined as

$$L := L(\epsilon_\theta) + \lambda L(\beta_\theta) \tag{16}$$

where λ is a coefficient that tradeoffs between the predicted Cauchy noise and squared scale.

Sampling Approach of Cauchy Diffusion

A stochastic sampling approach for the Cauchy Diffusion model is preferred because it provides better diversity than the deterministic sampling approach. Researchers have proposed various stochastic sampling approaches for Gaussian DDPMs (Nichol and Dhariwal 2021; Croitoru et al. 2023). However, these sampling approaches can not be applied directly to the Cauchy Diffusion model since they require estimating the posterior location, which the Cauchy Diffusion model does not support. Fortunately, we empirically found that the Cauchy Diffusion model supports deterministic and stochastic sampling with the methodology proposed in (Song, Meng, and Ermon 2020). From a higher-level perspective, this sampling methodology comprises three components: "predicted $\mathbf{x}_0$", "direction pointing to $\mathbf{x}_t$", and "random noise". Although the derivations are based on the Gaussian diffusion process, these components suit the Cauchy Diffusion model, possibly due to the similarities between the Cauchy and the Gaussian diffusion processes.

As the training objective of DDPMs relies on the marginals $q(\mathbf{x}_t|\mathbf{x}_0)$ instead of the distribution$q(\mathbf{x}_{1:T}|\mathbf{x}_0)$, a generalization of DDPMs using a class of non-Markovian diffusion processes has been proposed while maintaining the training objective unchanged (Song, Meng, and Ermon 2020). Since the Cauchy Diffusion model predicts the added Cauchy noises and their corresponding scales, the sampling process can be defined as:

$$\mathbf{x}_{t-1} = \sqrt{\bar{\alpha}_{t-1}}\left(\frac{\mathbf{x}_t - \sqrt{1-\bar{\alpha}_t}\epsilon_\theta}{\sqrt{\bar{\alpha}_t}}\right) + \sqrt{1-\bar{\alpha}_t-\eta\beta_\theta}\epsilon_\theta + \sqrt{\eta\beta_\theta}\epsilon_t \tag{17}$$

where ϵ_θ and β_θ denote the predicted Cauchy noise and the squared scale, respectively. A scaling factor η is adopted to determine the usage of the predicted squared scale during the sampling procedure. The deterministic sampling and stochastic sampling are applied when η equals to 0 and 1, respectively.

Experiments

Dataset and Evaluation Protocol

Dataset For fair comparison and better reproducibility, we used the LJSpeech dataset (Ito and Johnson 2017) and the VCTK dataset (Veaux, Yamagishi, and MacDonald 2017) to evaluate the speech synthesis performance of the Cauchy Diffusion model. The **LJSpeech** dataset consists of 13,100 audio files spoken by a female speaker. Each file is a single-channel 16-bit PCM WAV file with a sample rate of 22,050 Hz. The total duration of the dataset is approximately 24 hours, and the average duration of each audio file is about 6.5 seconds. We randomly selected 100 audio files from the dataset for performance evaluation and employed 13,000 audio files as the training set. The **VCTK** dataset includes 44455 audio files spoken by 110 English speakers. Each file is a single-channel 16-bit FLAC with a sample rate of 48000 Hz. The total duration of this dataset is about 44 hours, and the average duration of audio files is about 3.5 seconds. We employed the audio files of 105 speakers as the training set and randomly selected 20 audio files from each unseen speaker (5 speakers) for performance evaluation.

Evaluation Protocol Several objective and subjective metrics were applied to compare performance. We adopt several objective metrics for performance comparison, including the perceptual evaluation of speech quality (**PESQ**), short-time objective intelligibility (**STOI**), and Mel cepstral distortion (**MCD**) (Lam et al. 2022; Huang et al. 2022). We crowd-sourced 5-scale MOS tests via Amazon Mechanical Turk to obtain the mean opinion score (**MOS**) as a subjective metric for performance comparison (Chiang, Huang, and Lee 2023). The generated audio files were normalized to a loudness of -23 LUFS, and evaluated one at a time. Each raster evaluated the subjective naturalness of a sentence on a 1-5 Likert scale in 4 minutes. The definitions of scales are: *1:Bad - Completely unnatural speech; 2:Poor - Mostly unnatural speech; 3:Fair - Equally natural and unnatural speech; 4:Good - Mostly natural speech; 5:Excellent - Completely natural speech.* Five raters rated each audio file, and the MOS scores were computed with 95% confidence intervals (CI). The qualifications of the raters include a location in the US or GB, HIT Approval Rate $\geq$ 95%, and Number of HITs Approved $\geq$ 1000. Detailed instructions, including examples of different Likert scales, are provided to the raters. The MOS tests are conducted separately on the LJSpeech and VCTK datasets, where each test comprises 1000 audio files (900 files of 9 synthesized models and 100 files of ground truth). We paid \$0.05 for each rating, and the MOS tests on two datasets cost \$500.

Implementation Details

Data Preparation We employed the short-time Fourier transform (STFT) method proposed in (Wang et al. 2017) to generate the mel-spectrograms as conditions for speech synthesis. We used an 80-band mel-spectrogram as the conditions generated using the following STFT process settings. The FFT length, the hop length, and the window length are set to 1024, 256, and 1024, respectively. The minimal and maximal frequencies of the mel-spectrogram are set to 0 and 8000. In the training phase, we cropped 62 frames from the intact mel-spectrogram, corresponding to 15872 discretized points in the waveform.

Network Architecture In our proposed Cauchy Diffusion model, we employed a denoising neural network with a U-

Method	Setting	PESQ (↑)	STOI (↑)	MCD (↓)
Cauchy Diffusion ($\eta = 1, \text{NCV} = 5$)	$\lambda = 0.001$	4.013 ± 0.082	0.984 ± 0.005	1.938 ± 0.477
Cauchy Diffusion ($\eta = 1, \text{NCV} = 5$)	$\lambda = 0.1$	$\mathbf{4.019 \pm 0.072}$	0.984 ± 0.005	1.936 ± 0.481
Cauchy Diffusion ($\eta = 1, \text{NCV} = 5$)	$\lambda = 10$	4.014 ± 0.110	$\mathbf{0.985 \pm 0.005}$	$\mathbf{1.929 \pm 0.461}$
Cauchy Diffusion ($\eta = 1, \text{NCV} = 5$)	$\lambda = 1000$	4.018 ± 0.080	0.984 ± 0.006	1.939 ± 0.489
DiffWave (Kong et al. 2020)	$T = 50$	3.866 ± 0.118	0.978 ± 0.008	2.062 ± 0.521
Cauchy Diffusion ($\eta = 1, \text{NCV} = 5, \lambda = 10$)	$T = 50$	$\mathbf{3.994 \pm 0.110}$	$\mathbf{0.984 \pm 0.006}$	$\mathbf{2.021 \pm 0.521}$
FastDiff (Huang et al. 2022)	$T = 1000$	3.969 ± 0.096	0.980 ± 0.006	2.899 ± 0.762
Cauchy Diffusion ($\eta = 0, \text{NCV} = 5, \lambda = 10$)	$T = 1000$	$\mathbf{3.978 \pm 0.100}$	$\mathbf{0.982 \pm 0.006}$	$\mathbf{2.027 \pm 0.501}$
Cauchy Diffusion ($\eta = 0, \text{NCV} = 5, \lambda = 10, T = 1000$)	U-net1	3.978 ± 0.100	0.982 ± 0.006	2.027 ± 0.501
Cauchy Diffusion ($\eta = 0, \text{NCV} = 5, \lambda = 10, T = 1000$)	U-net2	$\mathbf{4.024 \pm 0.100}$	$\mathbf{0.983 \pm 0.005}$	$\mathbf{1.991 \pm 0.486}$
Cauchy Diffusion ($\eta = 1, \text{NCV} = 5, \lambda = 10, T = 1000$)	U-net1	4.014 ± 0.072	0.985 ± 0.005	1.929 ± 0.480
Cauchy Diffusion ($\eta = 1, \text{NCV} = 5, \lambda = 10, T = 1000$)	U-net2	$\mathbf{4.055 \pm 0.081}$	$\mathbf{0.986 \pm 0.005}$	$\mathbf{1.862 \pm 0.443}$

Table 1: Ablation studies of the tradeoff coefficients, diffusion steps, and model capacities on the LJSpeech dataset.

net structure (Ronneberger, Fischer, and Brox 2015). The network consists of a downsampling block and an upsampling block. A time embedding generated by a multi-layer feed-forward neural network is added to the raw waveform.

The **time embedding** is generated by a neural network of two layers, each with 512 units. The nonlinear activation function of these layers is the SiLU (Hendrycks and Gimpel 2016). The input to this neural network is the position encoding.

The **downsampling block** begins with a 1D convolution of kernel size 7 applied to the waveform, resulting in an output with 32 channels. The downsampling block comprises three modules with downsampling ratios of [4, 8, 8]. Three sequential 1D convolutions with a kernel size of 3 are employed within each downsampling module. The dilation factors and paddings of the 1D convolutions are set to [1, 2, 4]. Before the convolutions, the input to each downsampling module is projected to the desired dimensionality using the nearest interpolation. Subsequently, the 1D convolutions are performed on the projected feature. Additionally, a residual connection is established for each downsampling module. The residual connection involves a 1D linear convolution followed by nearest interpolation. Finally, the output of the last downsampling module and the residual output are summed together, producing the final output of the downsampling block.

The **upsampling block** in the model utilizes location-variable convolution (LVC) layers (Zeng et al. 2021). LVC layers are specifically designed to capture long-term dependencies of local acoustic features within speech waveforms. This is achieved by employing a kernel predictor, which generates multiple convolution kernels based on the mel-spectrogram. The upsampling block comprises three modules with upsampling ratios of [8, 8, 4]. Within each upsampling module, there are 4 sequential LVC layers. The hidden layer size for each LVC layer is set to 256. The kernel predictors' hidden channels and kernel size for each LVC layer are set to 64 and 3, respectively. Before applying the sequential LCV layers, the input to each upsampling module is first projected to the desired dimensionality using a transposed 1D convolution. A leaky ReLU (Xu et al. 2015) function is applied with a negative slope of 0.2 to introduce nonlinearity.

Training Routine All Cauchy Diffusion models were trained using the AdamW optimizer (Loshchilov and Hutter 2017) with a batch size of 64 and a learning rate of 2e-4. The optimizer's betas were set to (0.9, 0.98). Weight normalization (Salimans and Kingma 2016) was applied throughout the network. A gradient clip method constrained the maximum gradient norm to 1.0 is employed. Additionally, the exponential moving average (EMA) (Izmailov et al. 2018) technique with an update frequency of 10 steps and a ratio of 0.999 was employed to smooth the parameter updates. We trained all models on 4 NVIDIA 3090 GPUs, with a training speed of about eight steps per second.

Ablation Study

In this section, We evaluated the influence of difference hyperparameters for the Cauchy Diffusion model, including the Cauchy noise clamp value (NCV), the sampling style, the tradeoff coefficients, the diffusion step, and the model capacity. We conducted the ablation studies on the LJSpeech dataset. Unless otherwise specified, all models were trained for 10M steps with tradeoff coefficient set to 1000.

Noise Clamp Value The observation is that larger NCV leads to slower convergence. It can be seen from Figure 3 that the performances of all models improve as training progresses. However, the models' performances with larger NCVs remain consistently inferior to those with smaller NCVs. The results suggest that clamping the value of the Cauchy noise is necessary for faster convergence.

Sampling Style Stochastic sampling is expected to outperform deterministic sampling because it provides more prosody diversities in the generated speeches. As shown in Figure 3, stochastic sampling outperforms deterministic sampling when the NCV is 5. However, deterministic sampling outperforms stochastic sampling when the NCVs are 10 and 15. The underlying reason may be that the models with larger NCVs are converging slower.

Tradeoff Coefficient Since the Cauchy Diffusion model fixes the location to zero during the KL divergence computation, a larger coefficient should be adopted to promote

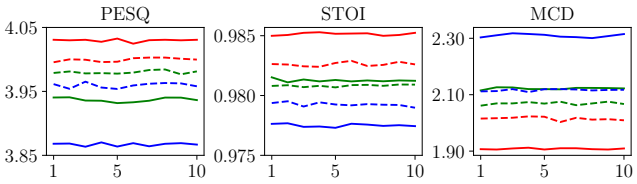

Figure 3: Ablation studies of the NCVs and sampling styles. The solid and dashed lines represent stochastic sampling and deterministic sampling, respectively. The red, green, and blue lines represent NCV=5, NCV=10, and NCV=15, respectively.

gradient flow from predicted scales to improve diversity. The results of the four tradeoff coefficients are shown in Table 1. As we can see, all models obtained similar quality (PESQ and STOI), but the model with the tradeoff coefficient setting to 10 achieved the best diversity (MCD). Note that the coefficient was set to 0.001 in (Nichol and Dhariwal 2021), though the Cauchy Diffusion model obtained its best diversity using a coefficient that is 1000 times larger.

Diffusion Step It is expected that longer diffusion steps lead to better performance for DDPMs (Nichol and Dhariwal 2021). We selected the DiffWave (Kong et al. 2020) using 50-step stochastic sampling and the FastDiff (Huang et al. 2022) using 1000-step deterministic sampling as baselines for comparison, and the results are shown in Table 1. As we can see, the Cauchy Diffusion model outperforms other methods at both diffusion steps. Also, from the performance listed in Table 1, we can infer that, compared with 50 diffusion steps, the Cauchy Diffusion model performed better with 1000 diffusion steps.

Model Capacity Two U-net configurations were adopted to generate Cauchy Diffusion models with different capacities. The first U-net is denoted as U-net1 and uses the described default configuration. The second U-net is denoted as U-net2 and uses the same configuration as U-net1, except that the upsampling blocks' dimensionalities are multiplied by 2, 4, and 8, respectively. The U-net1 and U-net2 models were trained for 10M and 3M steps, and the results are shown in Table 1. As we can see, the Cauchy Diffusion model with U-net2 obtained better performance with significantly fewer training steps. The results suggest that because of the heavy-tailed characteristics, we could increase the capacity of the denoising neural network for improved performance and reduced training steps.

Performance Comparison

Baselines We compared the performance of the Cauchy Diffusion model with several state-of-the-art neural vocoders, including a flow-based generative model of WaveGlow (Prenger, Valle, and Catanzaro 2019), generative adversarial network (GAN) based models of HiFiGAN (Kong, Kim, and Bae 2020) and UnivNet (Jang et al. 2021), and Gaussian DDPMs of WaveGrad (Chen et al. 2020), DiffWave (Kong et al. 2020), PriorGrad (Lee et al. 2022) and FastDiff (Huang et al. 2022). On both datasets, all models were trained with the same training audio files and evaluated on the unseen leave-out audio files. The Cauchy Diffusion models were built using the configuration U-net2 and trained for 3M steps with NCV set to 5. The WaveGlow and HifiGAN were trained for 2M steps, the UnivNet was trained for 2000 epochs, and all DDPMs were trained for 3M steps.

Objective Quality Comparison The comparison results between the Cauchy Diffusion model and other methods on the LJSpeech dataset and the VCTK dataset are presented in Table 2 and Table 3, respectively. The results demonstrate that the Cauchy Diffusion model obtained superior speech quality performances. On the LJSpeech dataset, the Cauchy Diffusion model obtained the best PESQ and STOI scores of 4.055 and 0.986, respectively. On the VCTK dataset, the Cauchy Diffusion model obtained the best PESQ score of 3.970 and an STOI score of 0.887, respectively. The STOI score obtained by the Cauchy Diffusion model on the VCTK dataset is worse than the STOI score of 0.960 obtained by the WaveGlow. However, the best STOI score obtained by DDPM-based neural vocoders is 0.941. The results may reflect that DDPMs face challenges obtaining the best STOI score on the VCTK dataset.

Objective Diversity Comparison The MCD score was taken for performance comparison of prosody diversity because rich prosody details can be embodied in the mel-spectrograms of the same sentence spoken by different person (Skerry-Ryan et al. 2018; Lee et al. 2021). The results are presented in Table 2 and Table 3. As we can see, the Cauchy Diffusion model achieved state-of-the-art prosody diversity on the LJSpeech and VCTK datasets. The Cauchy Diffusion model obtained the best MCD scores of 1.862 and 1.421 on the LJSpeech and VCTK datasets, respectively. Overall, the HiFiGAN and UnivNet models outperformed other methods regarding the MCD score because they designed the training objectives to optimize for better diversity (Kong, Kim, and Bae 2020; Jang et al. 2021). It is worth noting that the Cauchy Diffusion model beats other neural vocoders in prosody diversity with deterministic sampling, while stochastic sampling provides even better diversity. These results demonstrate the powerful ability of the Cauchy Diffusion model to capture prosody diversities and the necessity of supporting stochastic sampling.

Subjective Comparison We conducted the subjective comparison through crowd-sourced listening tests on the Amazon Mechanical Turk platform. The results are presented in the last columns of Table 2 and Table 3. Our Cauchy Diffusion model obtained the best MOS score on the LJSpeech dataset and a comparable MOS score with other DDPMs on the VCTK dataset. On the LJSpeech dataset, the Cauchy Diffusion model obtained the best MOS score of 4.00, the flow-based neural vocoder WaveGlow performed worst, and the GAN-based and DDPM-based neural vocoders performed better. On the VCTK dataset, the Cauchy Diffusion model obtained a MOS score of 3.76, comparable to GAN-based and DDPM-based neural vocoders. Interestingly, the flow-based neural vocoder

Method	PESQ (↑)	STOI (↑)	MCD (↓)	MOS (↑)
WaveGlow (Prenger, Valle, and Catanzaro 2019)	3.517 ± 0.149	0.953 ± 0.011	3.178 ± 0.572	3.75 ± 0.089
HiFiGAN (Kong, Kim, and Bae 2020)	3.679 ± 0.212	0.980 ± 0.007	2.136 ± 0.504	3.92 ± 0.074
UnivNet (Jang et al. 2021)	3.663 ± 0.193	0.978 ± 0.008	2.249 ± 0.518	3.93 ± 0.072
WaveGrad (Chen et al. 2020)	3.732 ± 0.155	0.972 ± 0.009	2.295 ± 0.523	3.91 ± 0.070
DiffWave (Kong et al. 2020)	3.866 ± 0.118	0.978 ± 0.008	2.062 ± 0.521	3.92 ± 0.070
PriorGrad (Lee et al. 2022)	3.973 ± 0.103	0.984 ± 0.005	4.147 ± 2.661	3.89 ± 0.070
FastDiff (Huang et al. 2022)	3.969 ± 0.096	0.980 ± 0.006	2.899 ± 0.762	3.92 ± 0.071
Cauchy Diffusion ($\eta = 0$)	4.024 ± 0.100	0.983 ± 0.005	1.991 ± 0.486	3.96 ± 0.073
Cauchy Diffusion ($\eta = 1$)	$\mathbf{4.055 \pm 0.081}$	$\mathbf{0.986 \pm 0.005}$	$\mathbf{1.862 \pm 0.443}$	$\mathbf{4.00 \pm 0.070}$
Ground Truth	-	-	-	3.94 ± 0.071

Table 2: Performance comparison of different speech synthesis methods on the LJSpeech dataset.

Method	PESQ (↑)	STOI (↑)	MCD (↓)	MOS (↑)
WaveGlow (Prenger, Valle, and Catanzaro 2019)	3.889 ± 0.368	$\mathbf{0.960 \pm 0.035}$	1.685 ± 0.891	$\mathbf{3.85 \pm 0.070}$
HiFiGAN (Kong, Kim, and Bae 2020)	3.748 ± 0.363	0.949 ± 0.047	1.635 ± 0.726	3.80 ± 0.074
UnivNet (Jang et al. 2021)	3.860 ± 0.279	0.939 ± 0.063	1.520 ± 0.654	3.75 ± 0.070
WaveGrad (Chen et al. 2020)	3.645 ± 0.242	0.863 ± 0.164	1.791 ± 0.785	3.71 ± 0.071
DiffWave (Kong et al. 2020)	3.812 ± 0.212	0.851 ± 0.187	1.686 ± 0.765	3.77 ± 0.068
PriorGrad (Lee et al. 2022)	3.925 ± 0.307	0.941 ± 0.065	3.117 ± 1.830	3.73 ± 0.069
FastDiff (Huang et al. 2022)	3.932 ± 0.197	0.903 ± 0.119	2.576 ± 0.873	3.78 ± 0.072
Cauchy Diffusion ($\eta = 0$)	$\mathbf{3.970 \pm 0.210}$	0.887 ± 0.157	1.493 ± 0.616	3.71 ± 0.068
Cauchy Diffusion ($\eta = 1$)	3.861 ± 0.273	0.877 ± 0.166	$\mathbf{1.421 \pm 0.606}$	3.76 ± 0.072
Ground Truth	-	-	-	3.77 ± 0.069

Table 3: Performance comparison of different speech synthesis methods on the VCTK dataset.

WaveGlow model performed best on the VCTK dataset. Given that the objective performance of WaveGlow also improves significantly on the VCTK dataset, we conjecture that WaveGlow may overfit the LJSpeech dataset. In contrast, the Cauchy Diffusion could perform better through sufficient training to improve the convergence status.

Discussion

The WaveNet and succeeding works lay the foundation of deep neural vocoders, yet the slow inference speed impedes their application (van den Oord et al. 2016, 2018). Flow-based vocoders significantly improve the inference speed (Prenger, Valle, and Catanzaro 2019; Kim et al. 2019), while GAN-based vocoders enable real-time synthesis with superior quality (Kong, Kim, and Bae 2020; Jang et al. 2021). However, GAN-based models are susceptible to model-collapse issues, so DDPM-based vocoders are introduced for better sample diversity (Chen et al. 2020; Kong et al. 2020; Huang et al. 2022). Researchers have recently proposed universal vocoders for performing out-of-distribution (OOD) zero-shot or few-shot synthesis (Lee et al. 2023). We believe universal vocoders are promising for developing new speech brain-computer interfaces (Qi et al. 2019; Tan et al. 2024) where OOD performance is critical. We will investigate the possibility of other noise schedules and applications to other domains (i.e., computer vision and OOD scenario) of Cauchy Diffusion.

The major distinctions between our work and other heavy-tailed generative models are the motivation and the derivation (Deasy, Simidjievski, and Liò 2021; YOON et al. 2023; Shariatian, Simsekli, and Durmus 2024; Pandey et al. 2024). First, we focus on the imbalanced speech dataset for better prosody diversity modeling, and other works focus on imbalanced image datasets for better quality of rare objects. Second, we aim to devise a heavy-tailed DDPM with a form similar to the original Gaussian DDPM, which drives us to formulate the Cauchy diffusion process neatly using two Gaussian diffusion processes that follow Kolmogorov equations through ratio distribution. In contrast, the heavy-tailed score-matching and DDPM models need massive derivations depending on specific heavy-tailed noises.

Conclusion

Denoising diffusion probabilistic models (DDPMs) have recently shown remarkable performance for speech synthesis. In this study, we explored the possibility of utilizing the Cauchy distribution to alleviate the mode-collapse issues of DDPM-based neural vocoders. We proposed a heavy-tailed DDPM called Cauchy Diffusion that incorporates the Cauchy noises into the DDPM through ratio distribution. We have also proposed a viable stochastic sampling approach for the Cauchy Diffusion model. The experimental results on two speech synthesis datasets demonstrate the effectiveness of the proposed Cauchy Diffusion model. Our proposal obtained state-of-the-art prosody speech diversity and superior speech quality. The main limitation of our proposal is that the Cauchy Diffusion model does not support fast sampling, which is critical for real-life deployment and application.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (62276228), the Key Research and Development Program of Zhejiang (2023C03001), the Zhejiang Provincial Natural Science Foundation of China (LR24F020002), and the Fundamental Research Funds for the Central Universities.

References

Cao, H.; Tan, C.; Gao, Z.; Xu, Y.; Chen, G.; Heng, P.-A.; and Li, S. Z. 2024. A survey on generative diffusion models. *IEEE Transactions on Knowledge and Data Engineering*.

Chen, N.; Zhang, Y.; Zen, H.; Weiss, R. J.; Norouzi, M.; and Chan, W. 2020. Wavegrad: Estimating gradients for waveform generation. *arXiv preprint arXiv:2009.00713*.

Chen, T. 2023. On the importance of noise scheduling for diffusion models. *arXiv preprint arXiv:2301.10972*.

Chiang, C.-H.; Huang, W.-P.; and Lee, H.-y. 2023. Why we should report the details in subjective evaluation of TTS more rigorously. *arXiv preprint arXiv:2306.02044*.

Chyzak, F.; and Nielsen, F. 2019. A closed-form formula for the Kullback-Leibler divergence between Cauchy distributions. *arXiv preprint arXiv:1905.10965*.

Croitoru, F.-A.; Hondru, V.; Ionescu, R. T.; and Shah, M. 2023. Diffusion models in vision: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Deasy, J.; Simidjievski, N.; and Liò, P. 2021. Heavy-tailed denoising score matching. *CoRR*, abs/2112.09788.

Feller, W. 1991. *An introduction to probability theory and its applications, Volume 2*, volume 81. John Wiley & Sons.

Foss, S.; Korshunov, D.; Zachary, S.; et al. 2011. *An introduction to heavy-tailed and subexponential distributions*, volume 6. Springer.

Han, P.; Ye, C.; Zhou, J.; Zhang, J.; Hong, J.; and Li, X. 2024. Latent-based Diffusion Model for Long-tailed Recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2639–2648.

Hendrycks, D.; and Gimpel, K. 2016. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*.

Ho, J.; Jain, A.; and Abbeel, P. 2020. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33: 6840–6851.

Huang, R.; Lam, M. W.; Wang, J.; Su, D.; Yu, D.; Ren, Y.; and Zhao, Z. 2022. Fastdiff: A fast conditional diffusion model for high-quality speech synthesis. *arXiv preprint arXiv:2204.09934*.

Ito, K.; and Johnson, L. 2017. The LJ Speech Dataset. https://keithito.com/LJ-Speech-Dataset/.

Izmailov, P.; Podoprikhin, D.; Garipov, T.; Vetrov, D.; and Wilson, A. G. 2018. Averaging weights leads to wider optima and better generalization. *arXiv preprint arXiv:1803.05407*.

Jabri, A.; Fleet, D.; and Chen, T. 2022. Scalable adaptive computation for iterative generation. *arXiv preprint arXiv:2212.11972*.

Jang, W.; Lim, D.; Yoon, J.; Kim, B.; and Kim, J. 2021. Univnet: A neural vocoder with multi-resolution spectrogram discriminators for high-fidelity waveform generation. *arXiv preprint arXiv:2106.07889*.

Kim, S.; Lee, S.; Song, J.; Kim, J.; and Yoon, S. 2019. FloWaveNet : A Generative Flow for Raw Audio. In Chaudhuri, K.; and Salakhutdinov, R., eds., *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, 3370–3378. PMLR.

Kong, J.; Kim, J.; and Bae, J. 2020. Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis. *Advances in neural information processing systems*, 33: 17022–17033.

Kong, Z.; Ping, W.; Huang, J.; Zhao, K.; and Catanzaro, B. 2020. Diffwave: A versatile diffusion model for audio synthesis. *arXiv preprint arXiv:2009.09761*.

Lam, M. W.; Wang, J.; Su, D.; and Yu, D. 2022. BDDM: Bilateral denoising diffusion models for fast and high-quality speech synthesis. *arXiv preprint arXiv:2203.13508*.

Lee, S.; Kim, H.; Shin, C.; Tan, X.; Liu, C.; Meng, Q.; Qin, T.; Chen, W.; Yoon, S.; and Liu, T. 2022. PriorGrad: Improving Conditional Denoising Diffusion Models with Data-Dependent Adaptive Prior. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Lee, S.; Ping, W.; Ginsburg, B.; Catanzaro, B.; and Yoon, S. 2023. BigVGAN: A Universal Neural Vocoder with Large-Scale Training. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Lee, S.-H.; Kim, J.-H.; Chung, H.; and Lee, S.-W. 2021. Voicemixer: Adversarial voice style mixup. *Advances in Neural Information Processing Systems*, 34: 294–308.

Loshchilov, I.; and Hutter, F. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Marsaglia, G. 1965. Ratios of normal variables and ratios of sums of uniform variables. *Journal of the American Statistical Association*, 60(309): 193–204.

McPherson, M. J.; and McDermott, J. H. 2018. Diversity in pitch perception revealed by task dependence. *Nature human behaviour*, 2(1): 52–66.

Nichol, A. Q.; and Dhariwal, P. 2021. Improved denoising diffusion probabilistic models. In *International conference on machine learning*, 8162–8171. PMLR.

Ning, Y.; He, S.; Wu, Z.; Xing, C.; and Zhang, L.-J. 2019. A review of deep learning based speech synthesis. *Applied Sciences*, 9(19): 4050.

Nooteboom, S.; et al. 1997. The prosody of speech: melody and rhythm. *The handbook of phonetic sciences*, 5: 640–673.

Pandey, K.; Pathak, J.; Xu, Y.; Mandt, S.; Pritchard, M. S.; Vahdat, A.; and Mardani, M. 2024. Heavy-Tailed Diffusion Models. *CoRR*, abs/2410.14171.

Prenger, R.; Valle, R.; and Catanzaro, B. 2019. Waveglow: A flow-based generative network for speech synthesis.

In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 3617–3621. IEEE.

Qi, Y.; Liu, B.; Wang, Y.; and Pan, G. 2019. Dynamic Ensemble Modeling Approach to Nonstationary Neural Decoding in Brain-Computer Interfaces. In Wallach, H. M.; Larochelle, H.; Beygelzimer, A.; d'Alché-Buc, F.; Fox, E. B.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, 6087–6096.

Qi, Y.; Wang, Y.; Zheng, X.; and Wu, Z. 2014. Robust feature learning by stacked autoencoder with maximum correntropy criterion. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2014, Florence, Italy, May 4-9, 2014*, 6716–6720. IEEE.

Qin, Y.; Zheng, H.; Yao, J.; Zhou, M.; and Zhang, Y. 2023. Class-balancing diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 18434–18443.

Ronneberger, O.; Fischer, P.; and Brox, T. 2015. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, 234–241. Springer.

Salimans, T.; and Kingma, D. P. 2016. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. *Advances in neural information processing systems*, 29.

Shariatian, D.; Simsekli, U.; and Durmus, A. 2024. Denoising Lévy Probabilistic Models. *CoRR*, abs/2407.18609.

Skerry-Ryan, R.; Battenberg, E.; Xiao, Y.; Wang, Y.; Stanton, D.; Shor, J.; Weiss, R.; Clark, R.; and Saurous, R. A. 2018. Towards end-to-end prosody transfer for expressive speech synthesis with tacotron. In *international conference on machine learning*, 4693–4702. PMLR.

Song, J.; Meng, C.; and Ermon, S. 2020. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*.

Tan, X.; Lian, Q.; Zhu, J.; Zhang, J.; Wang, Y.; and Qi, Y. 2024. Effective phoneme decoding with hyperbolic neural networks for high-performance speech BCIs. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*.

van den Oord, A.; Dieleman, S.; Zen, H.; Simonyan, K.; Vinyals, O.; Graves, A.; Kalchbrenner, N.; Senior, A. W.; and Kavukcuoglu, K. 2016. WaveNet: A Generative Model for Raw Audio. In Black, A. W., ed., *The 9th ISCA Speech Synthesis Workshop, SSW 2016, Sunnyvale, CA, USA, September 13-15, 2016*, 125. ISCA.

van den Oord, A.; Li, Y.; Babuschkin, I.; Simonyan, K.; Vinyals, O.; Kavukcuoglu, K.; van den Driessche, G.; Lockhart, E.; Cobo, L. C.; Stimberg, F.; Casagrande, N.; Grewe, D.; Noury, S.; Dieleman, S.; Elsen, E.; Kalchbrenner, N.; Zen, H.; Graves, A.; King, H.; Walters, T.; Belov, D.; and Hassabis, D. 2018. Parallel WaveNet: Fast High-Fidelity Speech Synthesis. In Dy, J. G.; and Krause, A., eds., *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, 3915–3923. PMLR.

Veaux, C.; Yamagishi, J.; and MacDonald, K. 2017. CSTR VCTK Corpus: English Multi-speaker Corpus for CSTR Voice Cloning Toolkit.

Wang, Y.; Skerry-Ryan, R.; Stanton, D.; Wu, Y.; Weiss, R. J.; Jaitly, N.; Yang, Z.; Xiao, Y.; Chen, Z.; Bengio, S.; et al. 2017. Tacotron: Towards end-to-end speech synthesis. *arXiv preprint arXiv:1703.10135*.

Xu, B.; Wang, N.; Chen, T.; and Li, M. 2015. Empirical evaluation of rectified activations in convolutional network. *arXiv preprint arXiv:1505.00853*.

Yang, L.; Zhang, Z.; Song, Y.; Hong, S.; Xu, R.; Zhao, Y.; Zhang, W.; Cui, B.; and Yang, M.-H. 2023. Diffusion models: A comprehensive survey of methods and applications. *ACM Computing Surveys*, 56(4): 1–39.

YOON, E. B.; Park, K.; Kim, S.; and Lim, S. 2023. Score-based generative models with Lévy processes. *Advances in Neural Information Processing Systems*, 36: 40694–40707.

Zeng, Z.; Wang, J.; Cheng, N.; and Xiao, J. 2021. MelGlow: Efficient waveform generative network based on location-variable convolution. In *2021 IEEE Spoken Language Technology Workshop (SLT)*, 485–491. IEEE.