



Diffusion Generative Vocoder for Fullband Speech Synthesis Based on Weak Third-order SDE Solver

Hideyuki Tachibana^{1,2}, Muneyoshi Inahara¹, Mocho Go¹, Yotaro Katayama¹, Yotaro Watanabe¹

¹PKSHA Technology Inc., Hongo, Bunkyo City, Tokyo, Japan

²Asia University, Sakai, Musashino City, Tokyo, Japan

h_tachibana@pkshatech.com

Abstract

Diffusion generative models, which generate data by the time-reverse dynamics of diffusion processes, have attracted much attention recently, and have already been applied in the speech domain such as speech waveform synthesis. Diffusion generative models initially had the disadvantage of slow synthesis, but many fast samplers have been proposed and this disadvantage is being overcome. The authors have also proposed an efficient sampler based on a second-order approximation derived from the Itô-Taylor series, and have achieved some success. This study further examines the possibility of incorporating third-order terms and experimentally verifies that a vocoder using this method can synthesize high-fidelity fullband (48 kHz) speech signals faster than in real time. It is also shown that the method is applicable to the extension of speech bandwidth from wideband (16 kHz) to fullband (48 kHz).

Index Terms: neural vocoder, speech superresolution, score-based model, bandwidth extension, Langevin dynamics.

1. Introduction

Many speech applications, including text-to-speech, speech denoising, voice conversion, etc., require a module which is called a vocoder to synthesize a waveform from acoustic features. In recent years, a number of methods for waveform synthesis based on deep generative models have been proposed including autoregressive models (e.g. WaveNet [1], WaveRNN [2]), generative adversarial networks (e.g. MelGAN [3], Parallel-WaveGAN [4]), normalizing flows (e.g. WaveGlow [5]), hybrid architectures with traditional signal processing (e.g. LPC-Net [6], DDSP [7], PeriodNet [8]), and diffusion models (e.g. DiffWave [9], VoiceGrad [10], WaveGrad [11]).

Among these deep generation frameworks, the diffusion models (Fig. 1) have attracted much attention recently, and have achieved significant results in many areas, especially in the image domain [12–14]. Diffusion models can be interpreted as tremendously deep neural networks based on iterative application of the same module (an update step of a discretized SDE/ODE). It has been reported that the methods have advantages over GANs which have been dominant in this field for years. They also have the advantage of elegant formulations backed by the theory of nonequilibrium thermodynamics and stochastic process [15, 16].

Diffusion generative models were initially said to have a fatal flaw in that they require a huge amount of computation for synthesis; typically, a deep neural network (DNN) needed to be evaluated hundreds of times to synthesize a single piece of data. However, with the rapid progress of the studies, many methods have been proposed to reduce the number of DNN evaluations and enable fast synthesis [17–23], and this drawback is being overcome. As one method in this series of trends,

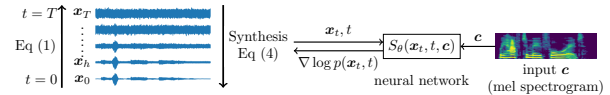


Figure 1: Concept of Diffusion Vocoder.

the authors proposed a method based on the Itô-Taylor expansion [23], and developed a neural vocoder *WaveGRIT* (WaveGrad + Itô-Taylor). The WaveGRIT has the advantage that, provided that some fundamental conditions are satisfied, it can synthesize a data faster and more accurately than a naïve sampler, using an existing pretrained checkpoint, without any additional fine-tuning nor distillation.

The purpose of this paper is to explore the further possibilities of the Itô-Taylor sampler. In particular, the following points will be examined and validated: (1) While we considered a second-order approximation in our previous paper [23], we derive a third-order approximation in this paper. Ideally, it is expected to be more accurate. We will confirm experimentally that this is actually the case in one aspect. (2) While most of existing neural vocoders have mainly dealt with the synthesis in medium sampling rates e.g. 22.05 kHz, this study experimentally verifies that the diffusion generative model is capable of synthesizing fullband signals with a sampling rate f_s of 48 kHz, and the training and synthesis are sufficiently efficient. (3) We also experimentally show that our method can upconvert speech signals from 16 kHz to 48 kHz. Such bandwidth extension techniques have been studied for many years [24–32], but little attempt has been made until recently [30, 32] to restore fullband speech signals containing the entire audible range.

2. Diffusion Generative Model

In this section, we consider a 1+1-dim case (1-dim x and 1-dim t) for simplicity. However, if the driving noise is diagonal, we may similarly consider the $l+1$ -dim case (l -dim x and 1-dim t) where l being the length of the signal to be synthesized.

2.1. Forward Noising Process

Let us first consider the following stochastic differential equation (SDE),

$$dx_t = -\frac{\beta_t}{2}x_t dt + \sqrt{\beta_t}dB_t, \quad (1)$$

where B_t is the Brownian motion. This class of SDEs are well studied and understood, and the solution of the above SDE is written as follows,

$$x_t = \sqrt{1 - \nu_t}x_0 + \sqrt{\nu_t}\varepsilon_t, \quad \text{where } \nu_t = 1 - e^{-\int_0^t \beta_s ds}. \quad (2)$$

The noise ε_t is Gaussian that satisfies $\mathbb{E}[\varepsilon_t] = 0$, $\mathbb{E}[\varepsilon_t^2] = 1$. This is obtained by considering the Fokker-Planck equation

$$\begin{aligned} \rho(t, h) &= 1 + \frac{\beta_t}{2}h + \frac{1}{4}\left(\frac{\beta_t^2}{2} - \dot{\beta}_t\right)h^2 + \frac{1}{48}(\beta_t^3 - 6\beta_t\dot{\beta}_t + 4\ddot{\beta}_t)h^3, \quad \mu(t, h) = -\frac{\beta_t}{\sqrt{\nu_t}}h + \frac{\dot{\beta}_t}{2\sqrt{\nu_t}}h^2 - \frac{1}{24\sqrt{\nu_t}}(\beta_t^3 + 4\ddot{\beta}_t)h^3 \\ \mathbf{n}(t, h) &= \sqrt{\beta_t}h\mathbf{w} - \left(\frac{(2-\nu_t)\beta_t^{3/2}}{2\nu_t}\mathbf{z} + \frac{\dot{\beta}_t}{2\sqrt{\beta_t}}(\mathbf{w}-\mathbf{z})\right)h^{3/2} - \frac{(-\nu_t^2 - 4\nu_t + 4)\beta_t^4 + 5\nu_t(\nu_t - 2)\beta_t^2\dot{\beta}_t - 2\nu_t^2\beta_t\ddot{\beta}_t + \nu_t^2\dot{\beta}_t^2}{24\nu_t^2\beta_t^{3/2}}\mathbf{w}h^{5/2} \end{aligned}$$

Figure 2: $\rho(t, h)$, $\mu(t, h)$ and $\mathbf{n}(t, h)$ in Eq (9)

(Kolmogorov’s forward equation) which describes the time evolution of the density function $p(x_t, t)$,

$$\frac{\partial}{\partial t}p(x_t, t) = -\frac{\partial}{\partial x_t}\frac{\beta_t}{2}p(x_t, t) + \frac{\partial^2}{\partial x_t^2}\frac{\beta_t}{2}p(x_t, t). \quad (3)$$

Assuming that the initial density is Dirac’s delta $p(x, 0) = \delta(x - x_0)$, the solution of this partial differential equation is written as $p(x_t, t | x_0, 0) = \mathcal{N}(x_t | \sqrt{1 - \nu_t}x_0, \nu_t)$, which is called the fundamental solution or the heat kernel. The unconditional density $p(x_t, t)$ is obtained by convolving the heat kernel with the initial distribution $p(x_0, 0)$. If β_t is designed so that $\nu_t \rightarrow 1$ at a large t , the density $p(x_t, t)$ goes to $\mathcal{N}(x_t | 0, 1)$.

2.2. Backward Denoising Process

In diffusion generative models, we are more interested in the following reverse-time SDE,

$$dx_t = \left(-\frac{\beta_t}{2}x_t - \beta_t\nabla \log p(x_t, t)\right)dt + \sqrt{\beta_t}dB_t. \quad (4)$$

The backward process is derived by considering Kolmogorov’s backward equation (KBE) and the Bayes theorem [33]. This backward SDE does not necessarily give the path-wise time-reversal of the forward SDE Eq (1). Instead, the dynamics of population, i.e. $p(x_t, t | x_T, T)$, ($t < T$) gives the inverse dynamics of the density evolution Eq (3). In other words, the backward SDE generates a data \hat{x}_0 that follows $p(x_0, 0 | x_T, T)$ from an initial Gaussian noise $x_T \sim p(x_T, T) = \mathcal{N}(x_T | 0, 1)$.

To solve the backward process above, we need the gradient of the log-probability term (score function). In the diffusion generative models, one designs it by a deep neural network $S_\theta(\mathbf{x}_t, t, \mathbf{c})$, and the network is trained so that the following approximate equality holds,

$$\begin{aligned} S_\theta(\mathbf{x}_t, t, \mathbf{c}) &\approx -\sqrt{\nu_t}\nabla \log p(\mathbf{x}_t, t) \approx -\sqrt{\nu_t}\nabla \log p(\mathbf{x}_t, t | \mathbf{x}_0, 0) \\ &= \frac{1}{\sqrt{\nu_t}}(\mathbf{x}_t - \sqrt{1 - \nu_t}\mathbf{x}_0) = \boldsymbol{\varepsilon}_t \end{aligned} \quad (5)$$

where \mathbf{c} is the conditioning information such as the mel-spectrogram. The reason multiplying the factor $-\sqrt{\nu_t}$ is to increase the trainability of the neural network by normalizing the variance of the training target (i.e. $\boldsymbol{\varepsilon}_t$) to be 1.

Here we want to achieve the approximate relation Eq (5) in each dimension. One of the simplest way for training such a network is to minimize the following L^1 score matching loss [11, 34], which can be considered to be a surrogate function for the negative ELBO [12],

$$Loss = \mathbb{E} \left[\|S_\theta(\sqrt{1 - \nu_t}\mathbf{x}_0 + \sqrt{\nu_t}\boldsymbol{\varepsilon}, t, \mathbf{c}) - \boldsymbol{\varepsilon}\|_1 \right] \quad (6)$$

where the expectation is taken w.r.t. and the training dataset $\mathcal{D} = \{(\mathbf{x}_0^{(i)} \in \mathbb{R}^l : \text{waveform}, \mathbf{c}^{(i)} : \text{mel-spec})\}_i$, the virtual time parameter $t \sim \mathcal{U}([0, T])$, and the Gaussian $\boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}_l, \mathbf{I}_l)$.

3. Proposed Method

3.1. Background of Itô-Taylor Sampler + Ideal Derivatives

To numerically solve the backward equation to generate a new sample, we need a discretized version of Eq (4). One of the simplest is the Euler-Maruyama (EM) approximation, obtained by replacing $dt \rightarrow h$, $d\bar{B}_t \rightarrow \sqrt{h}\mathbf{w}$, where $h > 0$ is the step size and $\mathbf{w} \sim \mathcal{N}(0, 1)$. The EM method is understood to be a first-order approximation based on the ‘‘Taylor expansion’’ of the stochastic system, but unlike the usual Taylor series for deterministic systems, the stochastic Taylor series must take into account the effect that the stochastic term is of the order of the square root ‘‘ $dB_t \sim \sqrt{dt}$.’’ This is solved by the famous Itô’s formula, and the modified Taylor series is called the Itô-Taylor series [35].

Since the EM method is essentially a first-order approximation, the step size h must be sufficiently small to accurately simulate the original SDE Eq (4); i.e., a large number of iterations are required. One idea to maintain the approximation accuracy even for a larger h is to incorporate higher order terms of the Itô-Taylor series. Such terms require the derivatives of network $S_\theta(\cdot, \cdot, \cdot)$. As a way to approximate these derivatives, the authors have proposed the *ideal derivative* model [23], in which the derivatives are written as follows,

$$\frac{\partial}{\partial \mathbf{x}}S_\theta(\mathbf{x}, t, \mathbf{c}) = \frac{1}{\sqrt{\nu_t}}, \quad (7)$$

$$\frac{\partial}{\partial t}S_\theta(\mathbf{x}, t, \mathbf{c}) = \frac{\beta_t}{2\sqrt{\nu_t}}\left(\mathbf{x} - \frac{S_\theta(\mathbf{x}, t, \mathbf{c})}{\sqrt{\nu_t}}\right). \quad (8)$$

The authors have also argued that any higher order numerical schemes based on the model is always written in a following form [23],

$$\mathbf{x}_{t-h} \leftarrow \rho(t, h)\mathbf{x}_t + \mu(t, h)S_\theta(\mathbf{x}_t, t, \mathbf{c}) + \mathbf{n}(t, h). \quad (9)$$

The scalar functions $\rho(t, h)$, $\mu(t, h)$ and the driving noise $\mathbf{n}(t, h)$ may be instantiated by the ones shown in Fig. 2. If we only use the **first order terms**, the sampling method is the conventional EM method mentioned. If we take into account the **second order terms**, we obtain a numerical scheme which is supposed to have the weak¹ convergence rate of 2.

3.2. Weak Third-order Itô-Taylor Sampler

Once we have confirmed that the second order approximation of Taylor series is largely successful [23], it is a natural next step to try to incorporate **third order terms**. In this paper, we adopted our idea of using the ideal derivatives to a weak third-order SDE solver based on the Itô-Taylor expansion, derived by Platen [35, § 14.3].

As the higher order terms of Itô-Taylor series are very complicated compared to the deterministic ones, we computed the

¹Roughly speaking, *weak convergence* is concerned with the accuracy of the moments of the ensemble, while *strong convergence* is concerned with the accuracy of individual paths.

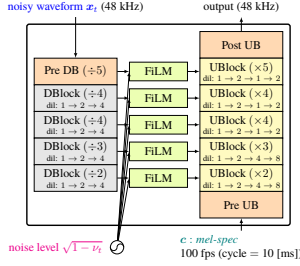


Figure 3: Diagram of the WaveGrad vocoder; up/down-sampling factors and dilation factors are modified. The number of parameters was 15.8M. For the details of the internal structure of the building blocks, see the original paper [11].

symbolic expressions of them using a computational algebra system SymPy [36]. Fig. 2 is the result of this symbolic calculation.

3.3. Driving Noise Options

The driving noise w, z that appear in $\mathbf{n}(t, h)$ should satisfy the conditions $\mathbb{E}[w] = \mathbb{E}[z] = \mathbf{0}_l, \mathbb{E}[ww^T] = \mathbf{I}_l, \mathbb{E}[zz^T] = \frac{1}{3}\mathbf{I}_l, \mathbb{E}[wz^T] = \frac{1}{2}\mathbf{I}_l$. These conditions can be satisfied by letting w, z as $w = \mathbf{u}_1, z = \frac{1}{2}\mathbf{u}_1 + \frac{1}{2\sqrt{3}}\mathbf{u}_2$, where $\mathbf{u}_1, \mathbf{u}_2 \sim \mathcal{N}(\mathbf{0}_l, \mathbf{I}_l)$. In the weak schemes of numerical SDEs, however, the random noise $\mathbf{u}_1, \mathbf{u}_2$ do not necessarily have to be white Gaussian noise if the conditions on means and covariances are satisfied. Indeed, as suggested in [35], we may also use the following *Binary noise* and *Ternary noise*,

$$\text{Binary: } p(\pm 1) = \frac{1}{2}, p(\text{otherwise}) = 0 \quad (10)$$

$$\text{Ternary: } p(\pm\sqrt{3}) = \frac{1}{6}, p(0) = \frac{2}{3}, p(\text{otherwise}) = 0 \quad (11)$$

as the conditions above are also satisfied when $\mathbf{u}_1, \mathbf{u}_2 \sim \text{Binary}(l)$, or $\mathbf{u}_1, \mathbf{u}_2 \sim \text{Ternary}(l)$.

Another option would be the *Purple noise*, which is obtained by differentiating and normalizing the Gaussian noise,

$$\text{Purple: } \tilde{v}_i := \frac{v_i - v_{i-1}}{\sqrt{2}}, \mathbf{v} \sim \mathcal{N}(\mathbf{0}_{l+1}, \mathbf{I}_{l+1}). \quad (12)$$

This noise does not satisfy the above conditions strictly but approximately. This noise is often used in the dithering process of digital audio production (i.e., adding noise on purpose to make quantization errors less noticeable), since it is said that the noise has less audible discomfort than quantization errors. We will also test this noise as a candidate of the driving noise.

3.4. Network Architecture

We used the same network structure as WaveGrad [11] except that the upscaling ($\times n$), downscaling ($\div n$) and dilation factors (dil) were modified which were essentially critical (Fig. 3). While the original WaveGrad upscaled the 80 fps mel-spectrogram by a factor of 300 to produce 24 kHz audio, our modified version upscales the 100 fps mel-spectrogram by a factor of 480 to produce 48 kHz audio.

4. Experiment

4.1. Configuration

■ **Dataset:** The dataset used for training was VCTK (48 kHz, 16 bit) [37]. Speech data from 98 of the 108 speakers were used

as the training set, and speech data from the other 10 speakers were used for evaluation.

■ **Specification of Input Mel-spectrogram:** We used the mel-spectrogram module provided in TorchAudio [38]. The window size and the FFT size when computing the base STFT were 42.6 [ms] (2048 points). The frame hop was 10 [ms] (480 points), meaning 100 frames per second. Each spectral frame was converted to mel-scale within the range of 80 Hz to 8 kHz. The number of mel-spectral bins was 80.

■ **Noise Schedule:** The noising schedule function β_t and ν_t were the same as the ones we derived previously in [23],

$$\beta_t = \dot{\lambda}_t \tanh \frac{\lambda_t}{2}, \nu_t = \tanh^2 \frac{\lambda_t}{2}, \lambda_t = \log(1 + Ae^{kt}). \quad (13)$$

The parameters A, k were determined so that the initial and terminal noise levels ν_0, ν_T equal to the given ones. In the training, the values were $\nu_0 = 10^{-6}, \nu_T = 0.999$, and $T = 1$.

■ **Training:** The speech signal was randomly cut with a length of 0.3 [s] (14,400 points) during training, and the resulting short segments were stacked to create a single mini-batch consisting of 96 segments. We trained the neural network model for 300k steps, using the Adam optimizer with default parameters except that the learning rate decayed exponentially by a factor of 0.9 per epoch. It took about 5 days on 4 GPUs (Tesla V100).

■ **Synthesis:** Starting from a Gaussian noise, $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}_l, \mathbf{I}_l)$, a speech signal \mathbf{x}_0 is synthesized by iteratively updating the waveform for T/h times using Eq (9). The initial and terminal noise levels were set $\nu_0 = 2 \times 10^{-7}, \nu_T = 0.999$. The total duration was $T = 1$ and the step size was $h = 0.02$, meaning that the number of refinement steps is $T/h = 50$. Under this condition, the real time factor (RTF) was about 0.7 for any cases. During synthesis, the sampling scheme was heuristically modified as follows. (1) The driving random noise was set to zero in the last 7 steps, i.e., $\mathbf{n}(t, h) = \mathbf{0}_l$. (2) After each refinement step, values outside the range $[-1, 1]$ were clipped.

■ **Comparative Method:** The baseline for comparison was WaveGlow [5] (94.4M params), which was a little modified to have the same input/output format as our method. Since the model size is 6 times larger than WaveGrad, the batch size was set to one-sixth, i.e. 16, due to memory limitations. The number of training steps was 130k. It also took about 5 days.

4.2. Waveform Synthesis from True Mel-spectrograms

Fig. 4 compares the spectrograms of the signals obtained by each sampling method from the mel-spectrograms excerpted from the aforementioned VCTK test set. It is visually confirmed that the third-order sampler has higher fidelity than the other samplers, especially in the high frequencies of consonants. Some audio samples are available at the author's website². This was also verified quantitatively by comparing the restoration error of log spectrograms for each band. Fig. 5(a) compares the band-wise mean log-spectral errors (L^2 distance), showing that the third-order scheme has a little higher fidelity than other methods, especially at high frequencies³. The worst log-spectral distance (L^p distance where $p \rightarrow \infty$) also showed a similar trend.

However, on the other hand, in our small-scale subjective evaluation (the crowdsourced MOS evaluation by participants with rewards), the difference was not clear (Table 1(a)). Since

²<https://tachi-hi.github.io/research/is2022>

³When evaluating the distance of the log-spectrograms, we clipped below -30 dB in order to prevent almost silent elements from affecting the results.

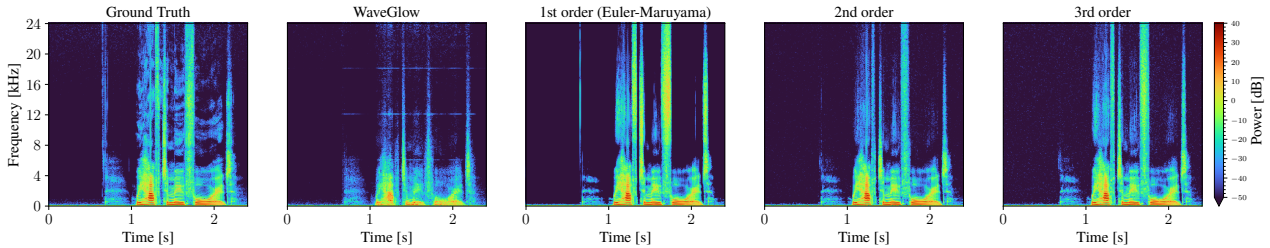


Figure 4: Examples of spectrograms of speech waveforms obtained by the WaveGlow, and the 1st, 2nd and 3rd order samplers.

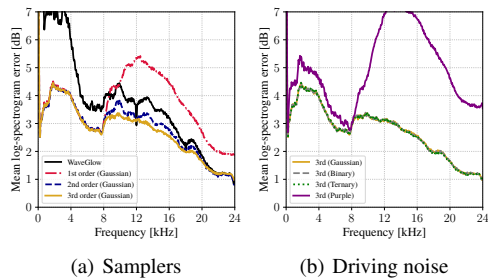


Figure 5: Mean log-spectrogram errors from ground truths (\downarrow).

Table 1: Mean Opinion Scores.
(a) Waveform Synthesis (b) Bandwidth Extension

Method	MOS (\uparrow)	Method	MOS (\uparrow)
Ground Truth	4.28 ± 0.80	Original (16k)	3.74 ± 0.87
WaveGlow	1.18 ± 0.38	Restored (48k)	2.94 ± 0.97
1st (EM)	3.78 ± 0.76	Hybrid	3.56 ± 0.94
2nd	3.66 ± 0.93		
3rd (Gaussian)	3.58 ± 0.85		
3rd (Purple)	3.24 ± 0.76		

high-frequency fidelity is not the only factor affecting the overall sound quality impression, the overall MOS was either not significantly different or slightly inferior for the higher-order schemes. In particular, since more driving noise is injected in higher-order schemes, it is possible that the residual noise lowered the MOS. Improvement of this point will be the subject of future study.

We also compared the effects of the driving noise differences. Qualitatively, contrary to our expectations, the purple noise did not reduce auditory discomfort due to its overemphasis on high frequency consonants, e.g. /s/, /f/, /t/, etc., and was inferior to others in all bands (Fig. 5(b)). On the other hand, other driving noises yielded almost the same results, from which we may conclude that the simplest one, viz. the binary noise, seems to be sufficient practically.

4.3. Wideband (16 kHz) - Fullband (48 kHz) Upconversion

As we have already seen, our method uses only below 8 kHz as input data, yet it can synthesize speech signals throughout all the bands up to 24 kHz. Therefore, this method can restore a fullband speech signal from a medium-quality speech data that contains only up to 8 kHz ($f_s \geq 16 \text{ kHz}$)⁴. Such a technique would be useful for converting old recordings to high resolution.

⁴Wideband audio does not always contain information up to 8 kHz, and cutoff frequencies as low as 7 kHz are sometimes used. We exclude such data in this paper, because the input format cannot meet our requirement that it contain information up to nearly 8 kHz.

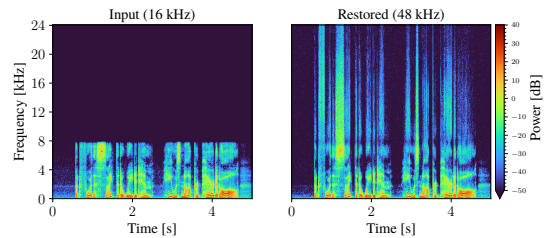


Figure 6: An example of speech bandwidth extension.

The specific procedure is as follows. First, a mel-spectrogram in a format compatible with our method is obtained from the 16 kHz input signal. Next, a waveform is synthesized from the mel-spectrogram using WaveGrad and our sampler. Additionally, since this method inevitably degrades the information in the original signal by converting it to the mel-spectrogram, it may be better to use the 16 kHz audio as it is, especially for low frequencies. Therefore, a method that uses the original audio below 8 kHz and the WaveGrad output above 8 kHz could be effective. We will call this a *hybrid method*.

Experiments were conducted to upconvert clips extracted from the LibriSpeech corpus [39] (16 kHz, 16 bit) using the above procedure. Fig. 6 shows a result of high-band restoration. Since the format of the input mel-spectrogram is the same and the domain of the task is similar to the previous section's one, the performance of high-band restoration is expected to be similar to Fig 5. Table 1(b) compares the MOS. Although the proposed method does not necessarily improve the listening impression due to the factors mentioned previously, the hybrid method does not seem to significantly degrade the MOS value.

5. Conclusion

In this paper, we proceeded from previous studies to further explore the possibilities regarding the application of diffusion generative models to vocoders. Specifically, we examined whether the performance can be improved by incorporating up to the third-order terms in the Itô-Taylor series, and the experimental results seemed positive. We also confirmed that the driving noise need not be Gaussian, but its covariance should be carefully considered. In addition, since the input mel-spectrogram is only required up to 8 kHz, our method can also be used for a bandwidth extension to upconvert 16 kHz audio to 48 kHz.

Research in this area has just begun and there are still many fundamental issues to consider, including the derivation of deterministic samplers [16, 17] and the consideration of classifier guidance [13]. Advancing the frontiers by accumulating knowledge on these fundamental issues and developing practical applications are the subjects of future study.

6. References

- [1] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, “WaveNet: A generative model for raw audio,” *arXiv:1609.03499*, 2016.
- [2] N. Kalchbrenner, E. Elsen, K. Simonyan, S. Noury, N. Casagrande, E. Lockhart, F. Stimberg, A. Oord, S. Dieleman, and K. Kavukcuoglu, “Efficient neural audio synthesis,” in *Proc. ICML*, 2018, pp. 2410–2419.
- [3] K. Kumar, R. Kumar, T. de Boissiere, L. Gestein, W. Z. Teoh, J. Sotelo, A. de Brébisson, Y. Bengio, and A. C. Courville, “MelGAN: Generative adversarial networks for conditional waveform synthesis,” *NeurIPS*, vol. 32, 2019.
- [4] R. Yamamoto, E. Song, and J.-M. Kim, “Parallel WaveGAN: A fast waveform generation model based on generative adversarial networks with multi-resolution spectrogram,” in *Proc. ICASSP*, 2020, pp. 6199–6203.
- [5] R. Prenger, R. Valle, and B. Catanzaro, “WaveGlow: A flow-based generative network for speech synthesis,” in *Proc. ICASSP*, 2019, pp. 3617–3621.
- [6] J.-M. Valin and J. Skoglund, “LPCNet: Improving neural speech synthesis through linear prediction,” in *Proc. ICASSP*, 2019, pp. 5891–5895.
- [7] J. Engel, L. Hantrakul, C. Gu, and A. Roberts, “DDSP: Differentiable digital signal processing,” in *Proc. ICLR*, 2020.
- [8] Y. Hono, S. Takaki, K. Hashimoto, K. Oura, Y. Nankaku, and K. Tokuda, “PeriodNet: A non-autoregressive waveform generation model with a structure separating periodic and aperiodic components,” in *Proc. ICASSP*, 2021, pp. 6049–6053.
- [9] Z. Kong, W. Ping, J. Huang, K. Zhao, and B. Catanzaro, “Diff-Wave: A versatile diffusion model for audio synthesis,” in *Proc. ICLR*, 2021.
- [10] H. Kameoka, T. Kaneko, K. Tanaka, N. Hojo, and S. Seki, “VoiceGrad: Non-parallel any-to-many voice conversion with annealed Langevin dynamics,” *arXiv:2010.02977*, 2020.
- [11] N. Chen, Y. Zhang, H. Zen, R. J. Weiss, M. Norouzi, and W. Chan, “WaveGrad: Estimating gradients for waveform generation,” in *Proc. ICLR*, 2020.
- [12] J. Ho, A. Jain, and P. Abbeel, “Denoising diffusion probabilistic models,” *NeurIPS*, vol. 33, pp. 6840–6851, 2020.
- [13] P. Dhariwal and A. Nichol, “Diffusion models beat GANs on image synthesis,” *NeurIPS*, vol. 34, 2021.
- [14] A. Nichol, P. Dhariwal, A. Ramesh, P. Shyam, P. Mishkin, B. McGrew, I. Sutskever, and M. Chen, “GLIDE: Towards photorealistic image generation and editing with text-guided diffusion models,” *arXiv:2112.10741*, 2021.
- [15] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli, “Deep unsupervised learning using nonequilibrium thermodynamics,” in *Proc. ICML*, 2015, pp. 2256–2265.
- [16] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, “Score-based generative modeling through stochastic differential equations,” in *Proc. ICLR*, 2020.
- [17] J. Song, C. Meng, and S. Ermon, “Denoising diffusion implicit models,” in *Proc. ICLR*, 2021.
- [18] Z. Kong and W. Ping, “On fast sampling of diffusion probabilistic models,” *arXiv:2106.00132*, 2021.
- [19] A. Jolicoeur-Martineau, K. Li, R. Piché-Taillefer, T. Kachman, and I. Mitliagkas, “Gotta go fast when generating data with score-based models,” *arXiv:2105.14080*, 2021.
- [20] D. Watson, J. Ho, M. Norouzi, and W. Chan, “Learning to efficiently sample from diffusion probabilistic models,” *arXiv:2106.03802*, 2021.
- [21] T. Salimans and J. Ho, “Progressive distillation for fast sampling of diffusion models,” in *Proc. ICLR*, 2022.
- [22] H. Zheng, P. He, W. Chen, and M. Zhou, “Truncated diffusion probabilistic models,” *arXiv:2202.09671*, 2022.
- [23] H. Tachibana, M. Go, M. Inahara, Y. Katayama, and Y. Watanabe, “Itô-Taylor sampling scheme for denoising diffusion probabilistic models based on ideal derivatives,” *arXiv:2112.13339*, 2021.
- [24] C. Avendano, H. Hermansky, and E. A. Wan, “Beyond Nyquist: towards the recovery of broad-bandwidth speech from narrow-bandwidth speech,” in *Proc. EUROSPEECH*, 1995, pp. 165–168.
- [25] D. Bansal, B. Raj, and P. Smaragdis, “Bandwidth expansion of narrowband speech using non-negative matrix factorization,” in *Proc. INTERSPEECH*, 2005, pp. 1505–1508.
- [26] S. Vaseghi, E. Zarehchi, and Q. Yan, “Speech bandwidth extension: Extrapolations of spectral envelop and harmonicity quality of excitation,” in *Proc. ICASSP*, 2006.
- [27] M. I. Mandel and Y. S. Cho, “Audio super-resolution using concatenative resynthesis,” in *Proc. WASPAA*, 2015.
- [28] V. Kuleshov, S. Z. Enam, and S. Ermon, “Audio super resolution using neural networks,” *arXiv:1708.00853*, 2017.
- [29] T. Y. Lim, R. A. Yeh, Y. Xu, M. N. Do, and M. Hasegawa-Johnson, “Time-frequency networks for audio super-resolution,” in *Proc. ICASSP*, 2018, pp. 646–650.
- [30] R. Kumar, K. Kumar, V. Anand, Y. Bengio, and A. Courville, “NU-GAN: High resolution neural upsampling with GAN,” *arXiv:2010.11362*, 2020.
- [31] Y. Li, M. Tagliasacchi, O. Rybakov, V. Ungureanu, and D. Roblek, “Real-time speech frequency bandwidth extension,” in *Proc. ICASSP*, 2021, pp. 691–695.
- [32] H. Liu, Q. Kong, Q. Tian, Y. Zhao, D. Wang, C. Huang, and Y. Wang, “VoiceFixer: Toward general speech restoration with neural vocoder,” *arXiv preprint arXiv:2109.13731*, 2021.
- [33] B. D. O. Anderson, “Reverse-time diffusion equation models,” *Stochastic Processes and their Applications*, vol. 12, no. 3, pp. 313–326, 1982.
- [34] A. Hyvärinen, “Estimation of non-normalized statistical models by score matching,” *Journal of Machine Learning Research*, vol. 6, no. 4, 2005.
- [35] P. E. Kloeden and E. Platen, *Numerical Solution of Stochastic differential Equations*. Springer, 1992.
- [36] A. Meurer, C. P. Smith, M. Paprocki, O. Čertík, S. B. Kirpichev, M. Rocklin, A. Kumar, S. Ivanov, J. K. Moore, S. Singh, T. Rathnayake, S. Vig, B. E. Granger, R. P. Muller, F. Bonazzi, H. Gupta, S. Vats, F. Johansson, F. Pedregosa, M. J. Curry, A. R. Terrel, Š. Roučka, A. Saboo, I. Fernando, S. Kulal, R. Cimrman, and A. Scopatz, “SymPy: symbolic computing in Python,” *PeerJ Computer Science*, vol. 3, p. e103, Jan. 2017.
- [37] C. Veaux, J. Yamagishi, and K. MacDonald, “Superseded-CSTR VCTK corpus: English multi-speaker corpus for CSTR voice cloning toolkit (version 0.92),” *University of Edinburgh. The Centre for Speech Technology Research (CSTR)*, 2019, DOI:10.7488/ds/2645.
- [38] Y.-Y. Yang, M. Hira, Z. Ni, A. Chourdia, A. Astafurov, C. Chen, C.-F. Yeh, C. Puhrsch, D. Pollack, D. Genzel, D. Greenberg, E. Z. Yang, J. Lian, J. Mahadeokar, J. Hwang, J. Chen, P. Goldsborough, P. Roy, S. Narenthiran, S. Watanabe, S. Chintala, V. Quenneville-Bélaïr, and Y. Shi, “TorchAudio: Building blocks for audio and speech processing,” *arXiv:2110.15018*, 2021.
- [39] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, “LibriSpeech: an ASR corpus based on public domain audio books,” in *Proc. ICASSP*, 2015, pp. 5206–5210.