FREGRAD: LIGHTWEIGHT AND FAST FREQUENCY-AWARE DIFFUSION VOCODER

Tan Dat Nguyen, Ji-Hoon Kim*, Youngjoon Jang, Jaehun Kim, Joon Son Chung*

Korea Advanced Institute of Science and Technology, South Korea

ABSTRACT

The goal of this paper is to generate realistic audio with a lightweight and fast diffusion-based vocoder, named FreGrad. Our framework consists of the following three key components: (1) We employ discrete wavelet transform that decomposes a complicated waveform into sub-band wavelets, which helps FreGrad to operate on a simple and concise feature space, (2) We design a frequency-aware dilated convolution that elevates frequency awareness, resulting in generating speech with accurate frequency information, and (3) We introduce a bag of tricks that boosts the generation quality of the proposed model. In our experiments, FreGrad achieves 3.7 times faster training time and 2.2 times faster inference speed compared to our baseline while reducing the model size by 0.6 times (only 1.78M parameters) without sacrificing the output quality. Audio samples are available at: https://mm.kaist.ac.kr/projects/FreGrad.

***Index Terms*—** speech synthesis, vocoder, lightweight model, diffusion, fast diffusion

1. INTRODUCTION

Neural vocoder aims to generate audible waveforms from intermediate acoustic features (e.g. mel-spectrogram). It becomes an essential building block of numerous speech-related tasks including singing voice synthesis [1, 2], voice conversion [3, 4], and text-to-speech [5, 6, 7]. Earlier neural vocoders [8, 9] are based on autoregressive (AR) architecture, demonstrating the ability to produce highly natural speech. However, their intrinsic architecture requires a substantial number of sequential operations, leading to an extremely slow inference speed. Numerous efforts in speeding up the inference process have been made on non-AR architecture based on flow [10, 11], generative adversarial networks [12, 13, 14], and signal processing [15, 16]. While such approaches have accelerated the inference speed, they frequently produce lower quality waveforms compared to AR methods. Among non-AR vocoders, diffusion-based vocoders have recently attracted increasing attention due to its promising generation quality [17, 18, 19, 20, 21, 22, 23]. Despite its high-quality synthetic speech, diffusion-based vocoder suffers from slow training convergence speed, inefficient inference process, and high computation cost. These factors hinder the utilization of diffusion-based vocoders in low-resource devices and their application in real-world scenarios. While many works [19, 21, 24] have tried to minimize training and inference times, there still remains a limited exploration to reduce computational costs.

To address the aforementioned problems at once, in this paper, we propose a novel diffusion-based vocoder called FreGrad, which achieves both low memory consumption and fast processing speed while maintaining the quality of the synthesized audio. The key to our idea is to decompose the complicated waveform into two simple frequency sub-band sequences (i.e. wavelet features), which allow our model to avoid heavy computation. To this end, we utilize discrete wavelet transform (DWT) that converts a complex waveform into two frequency-sparse and dimension-reduced wavelet features without a loss of information [25, 26]. FreGrad successfully reduces both the model parameters and denoise processing time by a significant margin. In addition, we introduce a new building block, named frequency-aware dilated convolution (Freq-DConv), which enhances the output quality. By incorporating DWT into the dilated convolutional layer, we provide the inductive bias of frequency information to the module, and thereby the model can learn accurate spectral distributions which serves as a key to realistic audio synthesis. To further enhance the quality, we design a prior distribution for each wavelet feature, incorporate noise transformation that replaces the sub-optimal noise schedule, and leverage a multi-resolution magnitude loss function that gives frequency-aware feedback.

In the experimental results, we demonstrate the effectiveness of FreGrad with extensive metrics. FreGrad demonstrates a notable enhancement in boosting model efficiency while keeping the generation quality. As shown in Table 1, FreGrad boosts inference time by 2.2 times and reduces the model size by 0.6 times with mean opinion score (MOS) comparable to existing works.

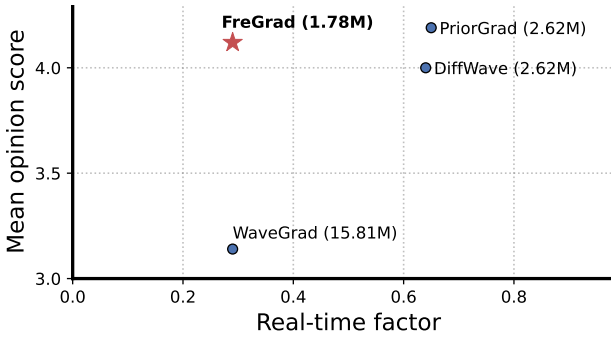


Fig. 1. FreGrad successfully reduces both real-time factor and the number of parameters while maintaining the synthetic quality.

2. BACKGROUNDS

The denoising diffusion probabilistic model is a latent variable model that learns a data distribution by denoising a noisy signal [27]. The *forward* process $q(\cdot)$ diffuses data samples through Gaussian transitions parameterized with a Markov process:

$$q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) = \mathcal{N}(\boldsymbol{x}_t; \sqrt{1-\beta_t}\boldsymbol{x}_{t-1}, \beta_t\mathbf{I}), \quad (1)$$

*These authors contributed equally to this work. This work was supported by the National Research Foundation of Korea grant funded by the Korean government (Ministry of Science and ICT, RS-2023-00212845) and the ITRC (Information Technology Research Center) support program (IITP-2024-RS-2023-00259991) supervised by the IITP (Institute for Information & Communications Technology Planning & Evaluation).

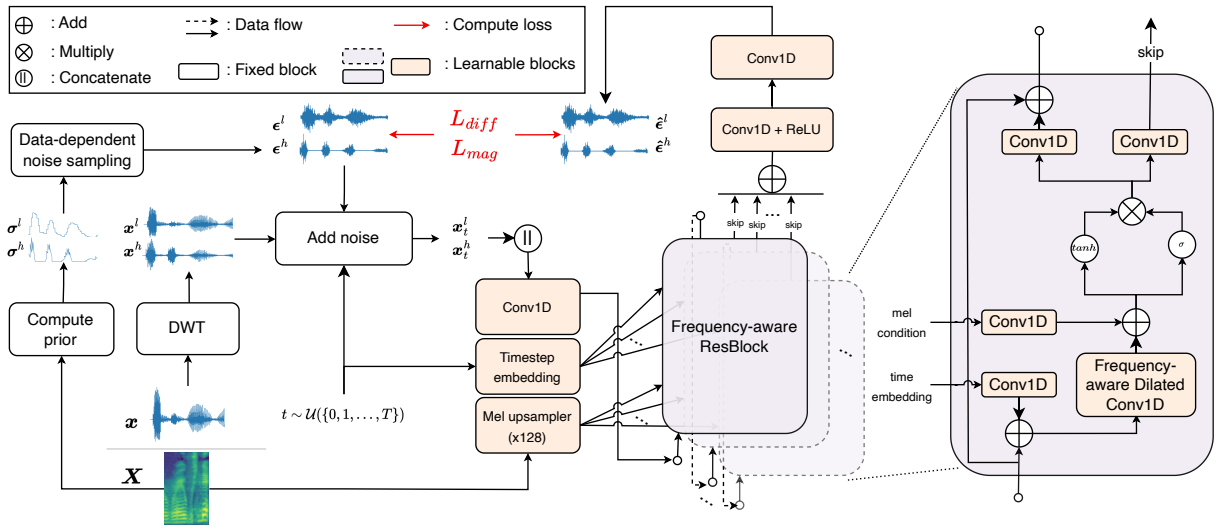


Fig. 2. Training procedure and model architecture of FreGrad. We compute wavelet features $\{\boldsymbol{x}^l, \boldsymbol{x}^h\}$ and prior distributions $\{\boldsymbol{\sigma}^l, \boldsymbol{\sigma}^h\}$ from waveform $\boldsymbol{x}$ and mel-spectrogram $\boldsymbol{X}$, respectively. At timestep t, noises $\{\boldsymbol{\epsilon}^l, \boldsymbol{\epsilon}^h\}$ are added to each wavelet feature. Given mel-spectrogram and timestep embedding, FreGrad approximates the noises $\{\hat{\boldsymbol{\epsilon}}^l, \hat{\boldsymbol{\epsilon}}^h\}$. The training objective is a weighted sum of $\mathcal{L}_{diff}$ and $\mathcal{L}_{mag}$ between ground truth and the predicted noise.

where $\beta_t \in \{\beta_1, \dots, \beta_T\}$ is the predefined noise schedule, T is the total number of timesteps, and $\boldsymbol{x}_0$ is the ground truth sample. This function allows sampling $\boldsymbol{x}_t$ from $\boldsymbol{x}_0$, which can be formulated as:

$$\boldsymbol{x}_t = \sqrt{\gamma_t}\boldsymbol{x}_0 + \sqrt{1-\gamma_t}\boldsymbol{\epsilon}, \tag{2}$$

where $\gamma_t = \prod_{i=1}^{t}(1-\beta_i)$ and $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.

With a sufficiently large T, the distribution of $\boldsymbol{x}_T$ approximates an Isotropic Gaussian distribution. Consequently, we can obtain a sample in ground truth distribution by tracing the exact *reverse* process $p(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)$ from an initial point $\boldsymbol{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. Since $p(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)$ depends on the entire data distribution, we approximate it with a neural network $p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)$ which is defined as $\mathcal{N}(\boldsymbol{x}_{t-1}; \mu_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t), \sigma^2_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t))$. As shown in [27], the variance $\sigma^2_{\boldsymbol{\theta}}(\cdot)$ can be represented as $\frac{1-\gamma_{t-1}}{1-\gamma_t}\beta_t$, and mean $\mu_{\boldsymbol{\theta}}(\cdot)$ is given by:

$$\mu_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t) = \frac{1}{\sqrt{1-\beta_t}}\left(\boldsymbol{x}_t - \frac{\beta_t}{\sqrt{1-\gamma_t}}\epsilon_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)\right), \tag{3}$$

where $\epsilon_{\boldsymbol{\theta}}(\cdot)$ is a neural network that learns to predict the noise.

In practice, the training objective for $\epsilon_{\boldsymbol{\theta}}(\cdot)$ is simplified to minimize $\mathbb{E}_{t,\boldsymbol{x}_t,\epsilon}\left[\|\boldsymbol{\epsilon} - \epsilon_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)\|_2^2\right]$. PriorGrad [20] extends the idea by starting the sampling procedure from the prior distribution $\mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$. Here, $\boldsymbol{\Sigma}$ is a diagonal matrix $diag\left[(\sigma_0^2, \sigma_1^2, \dots, \sigma_N^2)\right]$, where σ_i^2 is the i^{th} normalized frame-level energy of mel-spectrogram with length N. Accordingly, the loss function for $\epsilon_{\boldsymbol{\theta}}(\cdot)$ is modified as:

$$\mathcal{L}_{diff} = \mathbb{E}_{t,\boldsymbol{x}_t,\boldsymbol{\epsilon},\boldsymbol{c}}\left[\|\boldsymbol{\epsilon} - \epsilon_\theta(\boldsymbol{x}_t, t, \boldsymbol{X})\|^2_{\boldsymbol{\Sigma}^{-1}}\right], \tag{4}$$

where $\|\boldsymbol{x}\|^2_{\boldsymbol{\Sigma}^{-1}} = \boldsymbol{x}^\top\boldsymbol{\Sigma}^{-1}\boldsymbol{x}$ and $\boldsymbol{X}$ is a mel-spectrogram.

3. FREGRAD

The network architecture of FreGrad is rooted in DiffWave [17] which is a widely used backbone network for diffusion-based vocoders [20, 23]. However, our method is distinct in that it operates on a concise wavelet feature space and replaces the existing dilated convolution with the proposed Freq-DConv to reproduce accurate spectral distributions.

3.1. Wavelet Features Denoising

To avoid complex computation, we employ DWT before *forward* process. DWT downsamples the target dimension audio $\boldsymbol{x}_0 \in \mathbb{R}^L$ into two wavelet features $\{\boldsymbol{x}_0^l, \boldsymbol{x}_0^h\} \subset \mathbb{R}^{\frac{L}{2}}$, each of which represents low- and high-frequency components. As demonstrated in the previous works [26, 28], the function can deconstruct a non-stationary signal without information loss due to its biorthogonal property.

FreGrad operates on simple wavelet features. At each training step, the wavelet features $\boldsymbol{x}_0^l$ and $\boldsymbol{x}_0^h$ are diffused into noisy features at timestep t with distinct noise $\boldsymbol{\epsilon}^l$ and $\boldsymbol{\epsilon}^h$, and each noise is simultaneously approximated by a neural network $\epsilon_{\boldsymbol{\theta}}(\cdot)$. In *reverse* process, FreGrad simply generates denoised wavelet features, $\{\hat{\boldsymbol{x}}_0^l, \hat{\boldsymbol{x}}_0^h\} \subset \mathbb{R}^{\frac{L}{2}}$, which are finally converted into the target dimensional waveform $\hat{\boldsymbol{x}}_0 \in \mathbb{R}^L$ by inverse DWT (iDWT):

$$\hat{\boldsymbol{x}}_0 = \Phi^{-1}(\hat{\boldsymbol{x}}_0^l, \hat{\boldsymbol{x}}_0^h), \tag{5}$$

where $\Phi^{-1}(\cdot)$ denotes the iDWT function.

Note that FreGrad generates speech with smaller computations due to the decomposition of complex waveforms. In addition, the model maintains its synthetic quality, as iDWT guarantees a lossless reconstruction of a waveform from wavelet features [28, 29]. In our experiments, we adopt Haar wavelet [30].

3.2. Frequency-aware Dilated Convolution

Since audio is a complicated mixture of various frequencies [26], it is important to reconstruct accurate frequency distributions for natural audio synthesis. To enhance the synthetic quality, we propose Freq-DConv which deliberately guides the model to pay attention to the frequency information. As illustrated in Fig. 3, we adopt DWT to decompose the hidden signal $\boldsymbol{y} \in \mathbb{R}^{\frac{L}{2}\times D}$ into two sub-bands $\{\boldsymbol{y}_l, \boldsymbol{y}_h\} \subset \mathbb{R}^{\frac{L}{4}\times D}$ with hidden dimension D. The sub-bands are channel-wise concatenated, and the following dilated convolution $\mathbf{f}(\cdot)$ extracts a frequency-aware feature $\boldsymbol{y}_{hidden} \in \mathbb{R}^{\frac{L}{4}\times 2D}$:

$$\boldsymbol{y}_{hidden} = \mathbf{f}(\texttt{cat}(\boldsymbol{y}_l, \boldsymbol{y}_h)), \tag{6}$$

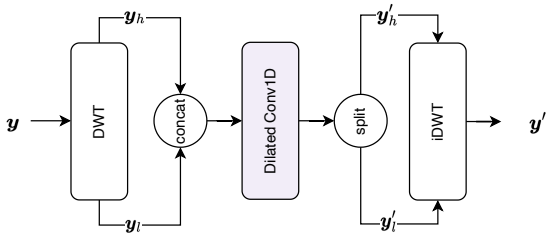


Fig. 3. Frequency-aware dilated convolution.

where `cat` denotes concatenation operation. The extracted feature $\boldsymbol{y}_{hidden}$ is then bisected into $\{\boldsymbol{y}'_l, \boldsymbol{y}'_h\} \subset \mathbb{R}^{\frac{L}{4} \times D}$ along channel dimension, and finally iDWT converts the abstract features into single hidden representation to match the length with input feature $\boldsymbol{y}$:

$$\boldsymbol{y}' = \Phi^{-1}(\boldsymbol{y}'_l, \boldsymbol{y}'_h), \tag{7}$$

where $\boldsymbol{y}' \in \mathbb{R}^{\frac{L}{2} \times D}$ represents the output of the Freq-DConv. As depicted in Fig. 2, we embed the Freq-DConv into every ResBlock.

The purpose of decomposing the hidden signal before the dilated convolution is to increase the receptive field along the time axis without changing the kernel size. As a result of DWT, each wavelet feature has a reduced temporal dimension while preserving all temporal correlations. This helps each convolution layer to possess a larger receptive field along the time dimension even with the same kernel size. Furthermore, low- and high-frequency sub-bands of each hidden feature can be explored separately. As a result, we can provide an inductive bias of frequency information to the model, which facilitates the generation of frequency-consistent waveform. We verify the effectiveness of Freq-DConv in Sec. 4.3.

3.3. Bag of Tricks for Quality

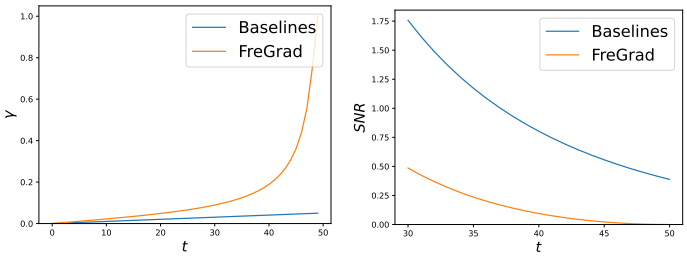
Prior distribution. As demonstrated in previous works [20, 22], a spectrogram-based prior distribution can significantly enhance the waveform denoising performance even with fewer sampling steps. Building upon this, we design a prior distribution for each wavelet sequence based on the mel-spectrogram. Since each sub-band sequence contains specific low- or high-frequency information, we use separate prior distribution for each wavelet feature. Specifically, we divide the mel-spectrogram into two segments along the frequency dimension and adopt the technique proposed in [20] to obtain separate prior distributions $\{\boldsymbol{\sigma}^l, \boldsymbol{\sigma}^h\}$ from each segment.

Noise schedule transformation. As discussed in [31, 32], signal-to-noise ratio (SNR) should ideally be zero at the final timestep T of *forward* process. However, noise schedules adopted in previous works [17, 18, 20] fail to reach SNR near zero at the final step, as shown in Fig. 4. To achieve a zero SNR at the final step, we adopt the proposed algorithm in [32], which can be formulated as follows:

$$\sqrt{\boldsymbol{\gamma}}_{new} = \frac{\sqrt{\gamma}_0}{\sqrt{\gamma}_0 - \sqrt{\gamma}_T + \tau}(\sqrt{\boldsymbol{\gamma}} - \sqrt{\gamma}_T + \tau), \tag{8}$$

where τ helps to avoid division by zero in sampling process.

Loss function. A common training objective of diffusion vocoder is to minimize the L2 norm between predicted and ground truth noise, which lacks explicit feedbacks in the frequency aspect. To give a frequency-aware feedback to the model, we add multi-resolution short-time Fourier transform (STFT) magnitude loss ($\mathcal{L}_{mag}$). Different from the previous works [14, 24], FreGrad only uses magnitude part since we empirically find that integrating *spectral convergence loss* downgrades the output quality. Let M be the number of STFT losses, then $\mathcal{L}_{mag}$ can be represented as:



(a) Noise level $\boldsymbol{\gamma}$ (b) SNR

Fig. 4. Noise level and log SNR through timesteps. "Baselines" refer to the work of [17, 18, 20] which use the same linear beta schedule $\boldsymbol{\beta}$ ranging from 0.0001 to 0.05 for 50 diffusion steps.

$$\mathcal{L}_{mag} = \frac{1}{M}\sum_{i=1}^{M} \mathcal{L}_{mag}^{(i)}, \tag{9}$$

where $\mathcal{L}_{mag}^{(i)}$ is STFT magnitude loss from i^{th} analysis settings [14]. We separately apply the diffusion loss to low- and high-frequency sub-bands, and the final training objective is defined as:

$$\mathcal{L}_{final} = \sum_{i \in \{l,h\}} \left[\mathcal{L}_{diff}(\boldsymbol{\epsilon}^i, \hat{\boldsymbol{\epsilon}}^i) + \lambda \mathcal{L}_{mag}(\boldsymbol{\epsilon}^i, \hat{\boldsymbol{\epsilon}}^i)\right], \tag{10}$$

where $\hat{\boldsymbol{\epsilon}}$ refers to an estimated noise.

4. EXPERIMENTS

4.1. Training Setup

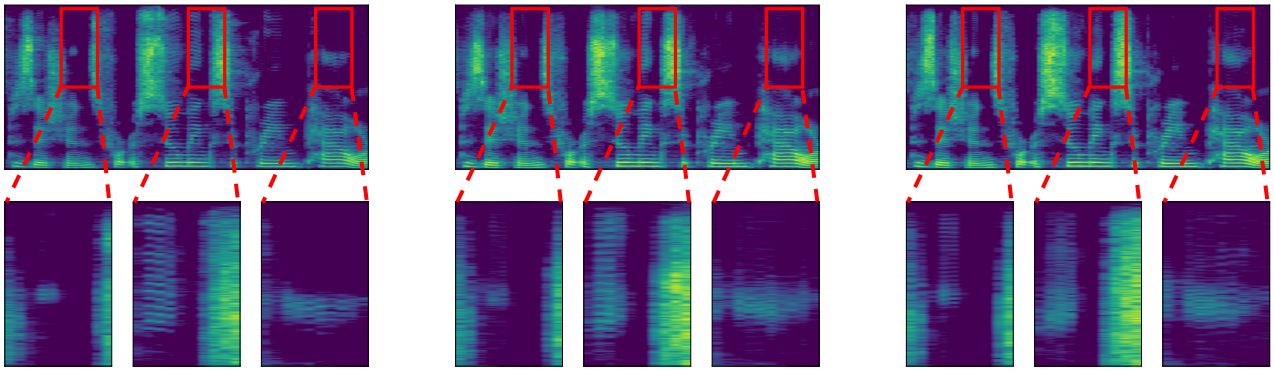
We conduct experiments on a single English speaker LJSpeech[1] which contains 13,100 samples. We use 13,000 random samples for training and 100 remaining samples for testing. Mel-spectrograms are computed from the ground truth audio with 80 mel filterbanks, 1,024 FFT points ranging from 80Hz to 8,000Hz, and hop length of 256. FreGrad is compared against the best performing publicly available diffusion vocoders: WaveGrad[2], DiffWave[3], and PriorGrad[4]. For fair comparison, all the models are trained until 1M steps, and all the audios are generated through 50 diffusion steps which is the default setting in DiffWave [17] and PriorGrad [20].

FreGrad consists of 30 frequency-aware residual blocks with a dilation cycle length of 7 and a hidden dimension of 32. We follow the implementation of DiffWave [17] for timestep embedding and mel upsampler but reduce the upsampling rate by half because the temporal length is halved by DWT. For $\mathcal{L}_{mag}$, we set $M = 3$ with FFT size of $[512, 1024, 2048]$ and window size of $[240, 600, 1200]$. We choose $\tau = 0.0001$ and $\lambda = 0.1$ for Eqn. (8) and Eqn. (10), respectively. We utilize Adam optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.999$, fixed learning rate of 0.0002, and batch size of 16.

[1] https://keithito.com/LJ-Speech-Dataset
[2] https://github.com/lmnt-com/wavegrad
[3] https://github.com/lmnt-com/diffwave
[4] https://github.com/microsoft/NeuralSpeech

Table 1. Evaluation results. The MOS results are presented with 95% confidence intervals. ↑ means higher is better, ↓ denotes lower is better.

Model	MOS ↑	MAE ↓	MR-STFT ↓	MCD_{13} ↓	RMSE_{f_0} ↓	#params ↓	RTF on CPU ↓	RTF on GPU ↓
Ground truth	4.74 ± 0.06	–	–	–	–	–	–	–
WaveGrad	3.14 ± 0.09	0.59	1.39	3.06	39.97	15.81M	**11.58**	**0.29**
DiffWave	4.00 ± 0.10	0.56	1.18	3.20	40.10	2.62M	29.99	0.64
PriorGrad	$\mathbf{4.19 \pm 0.10}$	0.47	1.14	2.22	40.42	2.62M	29.20	0.65
FreGrad	4.12 ± 0.11	**0.45**	**1.12**	**2.19**	**38.73**	**1.78M**	11.95	**0.29**



(a) Ground truth (b) FreGrad (c) PriorGrad

Fig. 5. Spectrogram analysis on FreGrad and PriorGrad. While PriorGrad suffers from over-smoothed results, FreGrad reproduces detailed spectral correlation, especially in red boxes.

4.2. Audio Quality and Sampling Speed

We verify the effectiveness of FreGrad on various metrics. To evaluate the audio quality, we obtain mel-cepstral distortion (MCD_{13}) and 5-scale MOS where 25 subjects rate the naturalness of 50 audio samples. In addition, we compute mean absolute error (MAE), $f0$ root mean square error (RMSE_{f0}), and multi-resolution STFT error (MR-STFT) between generated and ground truth audio. To compare the model efficiency, we calculate the number of model parameters (#params) and real-time factor (RTF) which is measured on AMD EPYC 7452 CPU and a single GeForce RTX 3080 GPU. Except for MOS, all the metrics are obtained from 100 audio samples.

As demonstrated in Table 1, FreGrad highly reduces not only the number of model parameters but also inference speed on both CPU and GPU. In addition, FreGrad achieves the best results in all the quality evaluation metrics except for MOS. Given humans' heightened sensitivity to low-frequency sounds, we hypothesize that the MOS degradation in FreGrad results from low-frequency distribution. However, in perspective of the entire spectrum of frequencies, FreGrad consistently demonstrates superior performance compared to existing methods, as confirmed by the MAE, MR-STFT, MCD_{13}, and RMSE_{f0}. The mel-spectrogram visualization analysis (Fig. 5) also demonstrates the effectiveness of FreGrad in reconstructing accurate frequency distributions. In addition, FreGrad takes significant advantage of fast training time. It requires 46 GPU hours to converge, 3.7 times faster than that of PriorGrad with 170 GPU hours.

4.3. Ablation Study on Proposed Components

To verify the effectiveness of each FreGrad component, we conduct ablation studies by using comparative MOS (CMOS), RMSE_{f0}, and RTF. In CMOS test, raters are asked to compare the quality of audio samples from two systems from -3 to $+3$. As can be shown in Table 2, each component independently contributes to enhancing the synthetic quality of FreGrad. Especially, the utilization of Freq-DConv substantially elevates the quality with a slight trade-off in inference speed, where the increased RTF still surpasses those of existing approaches. The generation qualities show relatively small but noticeable degradations when the proposed separate prior and zero SNR techniques are not applied. The absence of $\mathcal{L}_{mag}$ results in the worst performance in terms of RMSE_{f0}, which indicates that $\mathcal{L}_{mag}$ gives effective frequency-aware feedback.

Table 2. Ablation study for FreGrad components.

	CMOS ↑	RMSE_{f_0} ↓	RTF on GPU ↓
FreGrad	0.00	38.73	0.29
w/o Freq-DConv	-1.34	39.05	0.18
w/o separate prior	-0.26	38.91	0.29
w/o zero SNR	-0.69	39.17	0.29
w/o $\mathcal{L}_{mag}$	-0.68	39.82	0.29

5. CONCLUSION

We proposed FreGrad, a diffusion-based lightweight and fast vocoder. FreGrad operates on a simple and concise wavelet feature space by adopting a lossless decomposition method. Despite the small computational overhead, FreGrad can preserve its synthetic quality with the aid of Freq-DConv and the bag of tricks, which is designed specifically for diffusion-based vocoders. Extensive experiments demonstrate that FreGrad significantly improves model efficiency without degrading the output quality. Moreover, we verify the effectiveness of each FreGrad component by ablation studies. The efficacy of FreGrad enables the production of human-like audio even on edge devices with limited computational resources.

6. REFERENCES

[1] Jinglin Liu, Chengxi Li, Yi Ren, Feiyang Chen, and Zhou Zhao, "DiffSinger: Singing voice synthesis via shallow diffusion mechanism," in *Proc. AAAI*, 2022.

[2] Yi Ren, Xu Tan, Tao Qin, Jian Luan, Zhou Zhao, and Tie-Yan Liu, "DeepSinger: Singing voice synthesis with data mined from the web," in *Proc. KDD*, 2020.

[3] Kaizhi Qian, Yang Zhang, Shiyu Chang, Xuesong Yang, and Mark Hasegawa-Johnson, "AutoVC: Zero-shot voice style transfer with only autoencoder loss," in *Proc. ICML*, 2019.

[4] Hyeong-Seok Choi, Juheon Lee, Wansoo Kim, Jie Lee, Hoon Heo, and Kyogu Lee, "Neural analysis and synthesis: Reconstructing speech from self-supervised representations," in *NeurIPS*, 2021.

[5] Jonathan Shen, Ruoming Pang, Ron J. Weiss, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng Chen, Yu Zhang, Yuxuan Wang, RJ-Skerrv Ryan, Rif A. Saurous, Yannis Agiomyrgiannakis, and Yonghui Wu, "Natural TTS synthesis by conditioning wavenet on mel spectrogram predictions," in *Proc. ICASSP*, 2018.

[6] Vadim Popov, Ivan Vovk, Vladimir Gogoryan, Tasnima Sadekova, and Mikhail A. Kudinov, "Grad-TTS: A diffusion probabilistic model for text-to-speech," in *Proc. ICML*, 2021.

[7] Jaehyeon Kim, Sungwon Kim, Jungil Kong, and Sungroh Yoon, "Glow-TTS: A generative flow for text-to-speech via monotonic alignment search," in *NeurIPS*, 2020.

[8] Aäron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew W. Senior, and Koray Kavukcuoglu, "WaveNet: A generative model for raw audio," in *Proc. SSW*, 2016.

[9] Soroush Mehri, Kundan Kumar, Ishaan Gulrajani, Rithesh Kumar, Shubham Jain, Jose Sotelo, Aaron C. Courville, and Yoshua Bengio, "SampleRNN: An unconditional end-to-end neural audio generation model," in *Proc. ICLR*, 2017.

[10] Ryan Prenger, Rafael Valle, and Bryan Catanzaro, "WaveGlow: A flow-based generative network for speech synthesis," in *Proc. ICASSP*, 2019.

[11] Wei Ping, Kainan Peng, Kexin Zhao, and Zhao Song, "WaveFlow: A compact flow-based model for raw audio," in *Proc. ICML*, 2020.

[12] Kundan Kumar, Rithesh Kumar, Thibault de Boissiere, Lucas Gestin, Wei Zhen Teoh, Jose Sotelo, Alexandre de Brébisson, Yoshua Bengio, and Aaron C. Courville, "MelGAN: Generative adversarial networks for conditional waveform synthesis," in *NeurIPS*, 2019.

[13] Jesse H. Engel, Kumar Krishna Agrawal, Shuo Chen, Ishaan Gulrajani, Chris Donahue, and Adam Roberts, "GANSynth: Adversarial neural audio synthesis," in *Proc. ICLR*, 2019.

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

[15] Lauri Juvela, Bajibabu Bollepalli, Vassilis Tsiaras, and Paavo Alku, "GlotNet - A raw waveform model for the glottal excitation in statistical parametric speech synthesis," *IEEE/ACM Trans. on Audio, Speech, and Language Processing*, vol. 27, no. 6, pp. 1019–1030, 2019.

[16] Takuhiro Kaneko, Kou Tanaka, Hirokazu Kameoka, and Shogo Seki, "iSTFTNET: Fast and lightweight mel-spectrogram vocoder incorporating inverse short-time fourier transform," in *Proc. ICASSP*, 2022.

[17] Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro, "DiffWave: A versatile diffusion model for audio synthesis," in *Proc. ICLR*, 2021.

[18] Nanxin Chen, Yu Zhang, Heiga Zen, Ron J. Weiss, Mohammad Norouzi, and William Chan, "WaveGrad: Estimating gradients for waveform generation," in *Proc. ICLR*, 2021.

[19] Rongjie Huang, Max W. Y. Lam, Jun Wang, Dan Su, Dong Yu, Yi Ren, and Zhou Zhao, "FastDiff: A fast conditional diffusion model for high-quality speech synthesis," in *Proc. IJCAI*, 2022.

[20] Sang-gil Lee, Heeseung Kim, Chaehun Shin, Xu Tan, Chang Liu, Qi Meng, Tao Qin, Wei Chen, Sungroh Yoon, and Tie-Yan Liu, "PriorGrad: Improving conditional denoising diffusion models with data-dependent adaptive prior," in *Proc. ICLR*, 2022.

[21] Max W. Y. Lam, Jun Wang, Dan Su, and Dong Yu, "BDDM: Bilateral denoising diffusion models for fast and high-quality speech synthesis," in *Proc. ICLR*, 2022.

[22] Yuma Koizumi, Heiga Zen, Kohei Yatabe, Nanxin Chen, and Michiel Bacchiani, "SpecGrad: Diffusion probabilistic model based neural vocoder with adaptive noise spectral shaping," in *Proc. Interspeech*, 2022.

[23] Naoya Takahashi, Mayank Kumar, Singh, and Yuki Mitsufuji, "Hierarchical diffusion models for singing voice neural vocoder," in *Proc. ICASSP*, 2023.

[24] Zehua Chen, Xu Tan, Ke Wang, Shifeng Pan, Danilo P. Mandic, Lei He, and Sheng Zhao, "InferGrad: Improving diffusion models for vocoder by considering inference in training," in *Proc. ICASSP*, 2022.

[25] Ingrid Daubechies, *Ten Lectures on Wavelets*, SIAM, 1992.

[26] Ji-Hoon Kim, Sang-Hoon Lee, Ji-Hyun Lee, and Seong-Whan Lee, "Fre-GAN: Adversarial frequency-consistent audio synthesis," in *Proc. Interspeech*, 2021.

[27] Jonathan Ho, Ajay Jain, and Pieter Abbeel, "Denoising diffusion probabilistic models," in *NeurIPS*, 2020.

[28] Sang-Hoon Lee, Ji-Hoon Kim, Kangeun Lee, and Seong-Whan Lee, "Fre-GAN 2: Fast and efficient frequency-consistent audio synthesis," in *Proc. ICASSP*, 2022.

[29] Julien Reichel, Gloria Menegaz, Marcus J Nadenau, and Murat Kunt, "Integer wavelet transform for embedded lossy to lossless image compression," *IEEE Trans. on Image Processing*, vol. 10, no. 3, pp. 383–392, 2001.

[30] Alfred Haar, *Zur theorie der orthogonalen funktionensysteme*, Georg-August-Universitat, Gottingen., 1909.

[31] Emiel Hoogeboom, Jonathan Heek, and Tim Salimans, "Simple diffusion: End-to-end diffusion for high resolution images," in *Proc. ICML*, 2023.

[32] Shanchuan Lin, Bingchen Liu, Jiashi Li, and Xiao Yang, "Common diffusion noise schedules and sample steps are flawed," in *Proc. WACV*, 2024.