WAVENEXT 2: CONVNEXT-BASED FAST NEURAL VOCODERS WITH RESIDUAL DENOISING AND SUB-MODELING FOR GAN AND DIFFUSION MODELS

Wangzixi Zhou⋆†, *Takuma Okamoto*†, *Yamato Ohtani*†, *Sakriani Sakti*⋆, *Hisashi Kawai*†

⋆ Nara Institute of Science and Technology, Japan
† National Institute of Information and Communications Technology, Japan

ABSTRACT

Most neural vocoders are limited to one type: either GAN or diffusion-based. While state-of-the-art models like Vocos and WaveNeXt use powerful ConvNeXt-based generators, they have only been used in GAN frameworks and have limited performance in multi-speaker settings. Moreover, diffusion models, despite training faster than GANs, have slow CPU inference. In this paper, we introduce WaveNeXt 2, a unified ConvNeXt-based framework compatible with both GAN and diffusion vocoders. Its core innovation is residual denoising and sub-modeling, where each sub-model progressively refines the waveform. Experimental results in the multi-speaker dataset demonstrate the effectiveness of our approach: (1) GAN-WaveNeXt 2 is much faster than HiFi-GAN and WaveFit, and (2) Diff-WaveNeXt 2 also delivers much faster inference and competitive synthesis quality compared with FastDiff with 4 steps. The Diff-WaveNeXt 2 is very training-efficient, training in only 32 hours, making it ideal for resource-constrained applications.

Index Terms— Vocoder, Diffusion, GAN, ConveNext, unified generator

1. INTRODUCTION

Neural vocoders have become fundamental components in modern speech synthesis systems, responsible for generating high-fidelity speech waveforms from acoustic features, such as mel-spectrograms. Compared with fast autoregressive (AR) models, non-AR models are stable and many models have been proposed. The fast and high-fidelity Neural vocoders are broadly categorized into two main architectures: generative adversarial network (GAN)- and denoising diffusion probabilistic model (DDPM)-based neural vocoders. Each approach offers distinct advantages and trade-offs in terms of synthesis quality, inference speed, and training complexity.

GAN-based models [1, 2, 3, 4, 5, 6, 7, 8, 9]introduce a generator-discriminator framework to produce realistic waveforms with low latency. However, they often require substantial computational resources and are prone to instability during training. For example, training HiFi-GAN for 2.5 million steps can take over 300 hours on dual V100 GPUs. To mitigate these challenges, recent approaches, such as SpecDiff [10], integrate diffusion components to stabilize GAN training, while WaveFit [11] replaces stochastic noise injection with a fixed-point strategy to guide the synthesis process more reliably. In contrast, diffusion-based neural vocoders [12, 13, 14, 15, 16, 17, 18, 19, 20] leverage an iterative denoising process to generate speech waveforms by reversing a noise diffusion process. These models tend to be easier to train and more robust in certain scenarios but typically suffer from slow inference and potential degradation in output quality due to their multi-step generation pipeline. To improve inference efficiency, several techniques have been proposed to reduce the number of denoising steps. For instance, noise-level limited sub-modeling [21] trains specialized sub-models for different noise ranges, enhancing prediction accuracy. BDDM [22] further accelerates inference by learning a compact noise schedule, enabling high-quality synthesis in as few as four steps.

While numerous methods have been proposed to accelerate inference speed, most fast neural vocoders are limited to either GAN or diffusion models, limiting flexibility in real-world applications. Inspired by ConvNeXt architectures originally developed for image processing [23], recent generator designs in speech synthesis have attracted attention for their architectural simplicity and computational efficiency. Vocos [8] uses ConvNeXt blocks to predict STFT spectra, then reconstructs the waveform with an inverse STFT (iSTFT) layer. WaveNeXt [9] improves on this by using a trainable linear projection to directly predict the waveform, which enhances quality while maintaining speed. However, these promising ConvNeXt-based generators have only been used in GAN-based frameworks. While they offer faster inference than models like HiFi-GAN, they still show limited performance in multi-speaker situations. This highlights the need for more versatile and robust solutions.

To realize fast and high-fidelity neural vocoders for GAN and diffusion models, we propose WaveNeXt 2, a unified ConvNeXt-based generator framework compatible with both diffusion and GAN vocoders. WaveNeXt 2 is the initial framework applicable to both GAN- and diffusion-based fast neural vocoders within a single architecture on a CPU.

- We introduce ConvNeXt-based residual denoising and sub-modeling, in which each sub-model gradually performs denoising at each time step in inference. This enables a single architecture to be effectively applied across both vocoder types.
- We achieve significant improvements in real-time factor (RTF): GAN-WaveNeXt 2 offers much faster inference with comparable quality to HiFi-GAN, WaveFit, and the original WaveNeXt, while Diff-WaveNeXt 2 achieves faster inference and competitive quality relative to FastDiff.

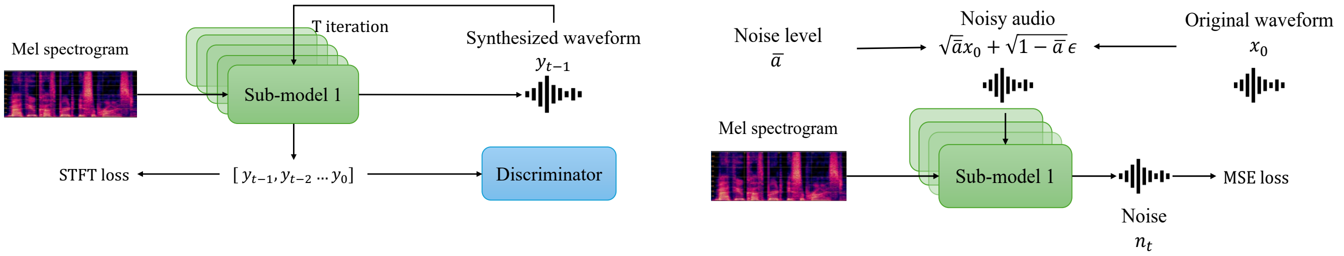
Speech samples from experiments are available on the demo page[1].

2. RELATED WORK

2.1. Non-AR Neural Vocoders

GAN-based neural vocoders: They use a generator and a discriminator in an adversarial process to produce high-quality speech. Models like HiFi-GAN [3] and MS-FC-HiFi-GAN [7] have improved

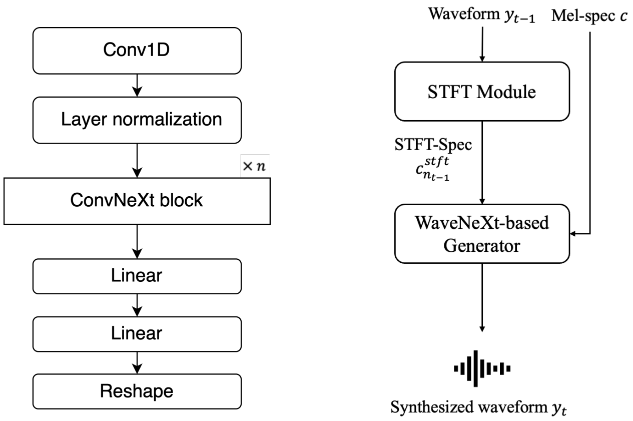
[1] https://37integer.github.io/WAVENEXT-2



(a) GAN-WaveNeXt 2

(b) Diff-WaveNeXt 2

Fig. 1: Training schemes of (a) GAN-WaveNeXt 2 and (b) Diff-WaveNeXt 2. In Diff-WaveNeXt 2, noise level $\overline{\alpha}$ is predefined with the noise schedule predictor BDDM.



(a) WaveNeXt-based generator

(b) Sub-model

Fig. 2: Overview of proposed architectures: (a) WaveNeXt-based generator where n is the number of ConvNeXt blocks. $n = 8$ is used in all the proposed models. (b) Sub-model for both GAN-WaveNeXt 2 and Diff-WaveNeXt 2, where t is the number of iterations steps.

synthesis quality and inference speed through specialized architectures. Despite these advancements, GANs still require significant computational resources and can suffer from training instability. While newer methods like SpecDiff [10] and WaveFit [11] attempt to improve stability, GAN vocoders often have slow CPU inference speeds.

Diffusion-based neural vocoders: They generate speech by reversing an iterative denoising process. The need for many steps to achieve high quality slows down inference. To counter this, techniques have been proposed to reduce the number of steps. FastDiff [16] uses specialized convolutions and a noise schedule predictor, while SpecGrad [15] employs an adaptive prior for better high-frequency quality. Additionally, noise-level limited sub-modeling [21] enhances prediction accuracy. However, even with these improvements, diffusion models still struggle with very few steps and their inference speed on CPUs remains slower than that of GANs.

2.2. ConvNeXt in neural vocoders

ConvNeXt [23], originally introduced in the image domain, demonstrated remarkable accuracy while maintaining architectural simplicity and computational efficiency . According to its outstanding performance, researchers have begun exploring its applications in the speech domain [8, 9, 24].

Vocos [8] integrates ConvNeXt layers into a neural vocoder. It predicts high-resolution STFT spectra from input mel-spectrograms, which are then converted into waveforms using an iSTFT layer. Vocos achieves inference speeds up to ten times faster than HiFi-GAN on a CPU.

WaveNeXt [9] further improves upon this by replacing the iSTFT layer in Vocos with a trainable linear projection layer that directly predicts waveform samples, eliminating the need for spectral representations. This modification preserves the fast inference speed of Vocos while enhancing speech quality.

3. PROPOSED APPROACH

We propose WaveNeXt 2, a unified framework that integrates ConvNeXt-based residual denoising sub-model into both GAN-based and diffusion-based architectures.

3.1. ConvNeXt-based residual sub-modeling

The architecture of the WaveNeXt-based generator is illustrated in Figure 2a. We retain the overall structure of the original WaveNeXt model [9], where the generator takes a mel-spectrogram as input and directly outputs the synthesized speech signal y_0. However, to enable a unified structure suitable for both GAN and diffusion frameworks, we modify the generator to predict the noise component n_t at each time step instead of generating the waveform directly. As shown in Fig. 2, the unified architecture comprises two main components: an STFT module and a WaveNeXt-based generator.

We first transform the input waveform y_{t-1} into its STFT representation using a Hann window. The STFT is computed with centering and produces a complex-valued spectrogram. The resulting complex spectrogram is then truncated along the temporal axis to match the duration of the target mel-spectrogram. The real and imaginary parts of the complex-valued STFT are separated for further processing. To form a real-valued spectral representation compatible with the mel-spectrogram input, we concatenate the full real part of the STFT with the imaginary part excluding the DC and Nyquist components (i.e., omitting the first and last frequency bins). This results in an STFT-spec, which along with the mel-spectrogram, is fed into the WaveNeXt-based generator to predict the noise component n_{t-1} at the current time step.

3.2. GAN-based model: GAN-WaveNeXt 2

The proposed GAN-WaveNeXt 2 is illustrated in Figure 1a. During training, we adopt the fixed-point iteration strategy introduced in WaveFit [25], which differs from conventional DDPMs by deterministically guiding each denoising step toward the target waveform, rather than relying on stochastic noise removal. The training process is structured as follows: In each iteration, the sub-model receives a

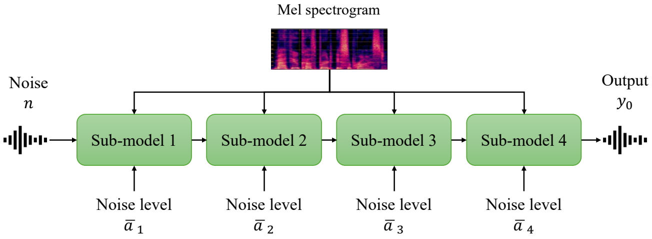


Fig. 3: Inference procedure of Diff-WaveNeXt 2 with 4 sub models for 4 iterations.

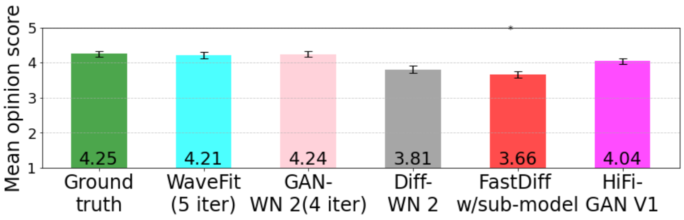


Fig. 4: Results of MOS tests with 20 listening subjects. The confidence level is 95%. WN means WaveNeXt.

mel-spectrogram and a noisy waveform y_t as input, and predicts the next denoised waveform y_{t-1}. This process is repeated for T steps until the final waveform y_0 is synthesized.

While our training pipeline is similar to WaveFit's, we have simplified it. WaveFit's original pipeline gradually converts initial input noise into clean speech based on fixed-point iteration, but we found that a "denoising" constraint is not necessary in the training loss, meaning initial input noise isn't required. We also omitted WaveFit's gain adjustment modules as they are redundant with the STFT loss. Preliminary experiments confirmed that GAN-WaveNeXt 2 performs effectively without either the initial input noise or the gain adjustment modules, allowing for a simplified training process.

3.3. Diffusion-based model: Diff-WaveNeXt 2

The architecture of the proposed Diff-WaveNeXt 2 is illustrated in Figure1b. Rather than following the original DDPM training strategy, we adopt the training strategy proposed in [21], in which each sub-model is trained separately, each responsible for denoising within a specific range of noise levels. To implement this, we divide the denoising task into four stages and construct four sub-models. Each sub-model is conditioned not only on the mel-spectrogram but also on a specific noisy audio, denoted by $x_t = \sqrt{\overline{\alpha_t}}x_0 + \sqrt{1-\overline{\alpha_t}}\epsilon$, where ϵ represents Gaussian noise, x_0 is original clean waveform and $\overline{\alpha_t}$ is the cumulative noise level predicted for step t .

The inference process is illustrated in Figure 3. The mel-spectrogram and the corresponding noise level $\overline{\alpha}$ are provided as inputs to the respective sub-models. Starting from an initial noise signal $\boldsymbol{n}$, the four sub-models are applied sequentially, each responsible for denoising within a specific noise level range. After four sub-model, the final output is the synthesized speech waveform $\boldsymbol{y}_0$. According to [21], the high-frequency details in synthesized speech are often lost when the noise schedule contains unnecessary noise, especially with a low number of iterations. To restore these missing components, we also use the time-invariant spectral enhancement post-filtering technique introduced in the same paper.

4. EXPERIMENTS

All the models were implemented using PyTorch [26] and trained on NVIDIA A100 GPUs with 40 GB of memory.

4.1. Experimental conditions

Dataset: We trained all the models on LibriTTS-R [27], which is a multi-speaker English corpus of approximately 585 hours of read English speech at 24 kHz sampling rate. We trained model from the combination of the "train-clean-100" and "train-clean-360" subsets at 24 kHz sampling.

Model and training setting: For GAN-based models, we introduced unofficial implementations of HiFi-GAN V1[2] and WaveFit[3]. For diffusion-based models, we introduced an official implementation of FastDiff[4]. To implement WaveNeXt-based generator, we introduced an official implementation of Vocos[5] and replaced the STFT layer with linear layer. For all the models, 128-dimensional mel-spectrograms were used as input acoustic features in WaveFit. To ensure a fair comparison with baseline systems, we matched the hop sizes of the proposed models to those of the corresponding baselines: GAN-WaveNeXt 2 and HiFi-GAN adopted a hop size of 300, consistent with WaveFit, while Diff-WaveNeXt 2 used a hop size of 256, matching that of FastDiff. In GAN-WaveNeXt 2, we employed the same discriminators and loss functions as WaveFit to ensure training stability and consistency across adversarial learning. All losses follow the same definitions as those in WaveFit [11]. In Diff-WaveNeXt 2, we trained four independent sub-models to handle different noise levels. The noise schedule used for training was predicted by a noise schedule predictor adapted from BDDM [22]. The resulting noise schedule with 4 steps was [1.0e−04, 2.8e−02, 5.6e−01, 9.1e−01].

Evaluation criteria: To evaluate the naturalness of the synthesized speech subjectively, we conducted mean opinion score (MOS) tests [28] using a five-point scale. A total of 20 paid native English speakers participated in the evaluation. All subjects use headphones in a quiet environment to listen. In total, each participant evaluated 120 samples (20 utterances × 6 models). In addition subjective evaluation, we employed objective evaluation methods to assess speech quality. We measured UTMOS [29] and NISQA [30], which are automatic MOS prediction models. And we adopted two widely used signal-based objective metrics: mel-cepstral distortion (MCD) [31] and log F0 root-mean-square error (RMSE), both of which provide quantitative assessments of spectral and prosodic accuracy. For inference speed, we measured RTFs on an NVIDIA A100 GPU and an AMD EPYC 7542 CPU (1 core).

Ablation study: For Diff-WaveNeXt 2, we also implemented our model within the original FastDiff architecture without submodeling. However, the performance was suboptimal compared to our unified framework. Therefore, we introduced sub-modeling train strategy [21] to realize better performance. The complete ablation results and comparisons are summarized in Table 1.

[2] https://github.com/kan-bayashi/ParallelWaveGAN
[3] https://github.com/yukara-ikemiya/wavefit-pytorch
[4] https://github.com/Rongjiehuang/FastDiff
[5] https://github.com/gemelo-ai/vocos

Table 1: Results of mel-cepstral distortion (MCD), log F0 root-mean-square error (RMSE), UTMOS and NISQA columns represent the means and standard deviations. Real-time factor (RTF) of the inference. Proposed methods are described in **bold**.

	RTF(GPU) ↓	RTF(CPU) ↓	NISQA ↑	UTMOS ↑	MCD ↓	log F0 RMSE ↓	Model size
Ground Truth	–	–	4.08 ± 0.19	4.11 ± 0.09	–	–	
WaveNeXt (1 iteration)	**0.0022**	**0.06**	3.16 ± 0.24	3.20 ± 0.12	0.92 ± 0.52	0.31 ± 0.15	14.98M
WaveFit (2 iterations)	0.0111	2.15	3.80 ± 0.22	3.89 ± 0.11	1.03 ± 0.54	0.32 ± 0.15	15.51M
GAN-WaveNeXt 2 (2 iterations)	0.0033	0.10	3.77 ± 0.20	3.88 ± 0.11	0.97 ± 0.54	0.31 ± 0.15	29.97M
WaveFit (3 iterations)	0.0151	3.22	3.91 ± 0.22	3.98 ± 0.10	1.01 ± 0.54	0.32 ± 0.13	15.51M
GAN-WaveNeXt 2 (3 iterations)	0.0054	0.15	3.92 ± 0.22	3.91 ± 0.10	0.96 ± 0.57	0.30 ± 0.18	44.96M
WaveFit (4 iterations)	0.0213	4.28	3.97 ± 0.21	3.99 ± 0.10	1.01 ± 0.52	0.32 ± 0.11	15.51M
GAN-WaveNeXt 2 (4 iterations)	0.0066	0.20	4.01 ± 0.20	4.04 ± 0.09	0.95 ± 0.53	0.30 ± 0.11	59.94M
WaveFit (5 iterations)	0.0226	5.36	4.02 ± 0.19	4.04 ± 0.09	**0.90 ± 0.52**	0.31 ± 0.13	15.51M
GAN-WaveNeXt 2 (5 iterations)	0.0090	0.24	4.01 ± 0.19	4.04 ± 0.09	0.95 ± 0.51	0.30 ± 0.12	74.93M
HiFi-GAN V1	0.0110	0.80	3.99 ± 0.22	**4.05 ± 0.11**	2.34 ± 0.83	0.16 ± 0. 01	13.9M
FastDiff wo/ sub-modeling	0.0625	0.80	3.43 ± 0.20	3.50 ± 0.11	4.76 ± 0. 74	0.16 ± 0. 01	15.63M
Diff-WaveNeXt 2 wo/ sub-modeling	0.0335	0.16	3.45 ± 0.19	3.55 ± 0.09	7.34 ± 1. 46	0.16 ± 0. 01	14.42M
FastDiff w/ sub-modeling	0.0282	0.80	3.67 ± 0.20	3.78 ± 0.06	4.32 ± 0.69	0.24 ± 0.33	62.52M
Diff-WaveNeXt 2	0.0164	0.16	3.81 ± 0.19	3.87 ± 0.05	4.16 ± 0. 88	**0. 12 ± 0. 01**	57.68M

Table 2: Training time of models in a single GPU

Model	Training time
GAN-WaveNeXt 2	410 hours
HiFi-GAN	270 hours
WaveFit	410 hours
Diff-WaveNeXt 2	**32 hours**
Fastdiff	96 hours

4.2. Results and discussion

We evaluated the performance of our proposed models in terms of both the inference speed and synthesized speech quality. The RTFs on a GPU and a CPU, as well as objective speech quality metrics—log F0 RMSE, Mel Cepstral Distortion (MCD), UTMOS, and NISQA—are shown in Table 1. The evaluations were conducted using 4,824 samples from the LibriTTS-R [27] "test-clean-100" subset. Additionally, the results of MOS tests obtained from 20 samples of the same subset are used to assess subjective quality. These are shown in Figure 4.

The results demonstrate that GAN-WaveNeXt 2 achieves UTMOS and NISQA scores comparable to those of WaveFit, while drastically improving the inference speed. Specifically, GAN-WaveNeXt 2 achieves a 70% reduction in RTF on GPU and a 90% reduction on CPU, compared to WaveFit. As illustrated in Figure 4, GAN-WaveNeXt 2 with 4 iterations achieves comparable MOS scores with both WaveFit with 5 iterations and HiFi-GAN, while also surpassing HiFi-GAN in terms of the inference speed—offering a 40% improvement on GPU and 75% on CPU. Although GAN-WaveNeXt 2 performs higher log F0 RMSE compared to HiFi-GAN, it performs better in terms of MCD, indicating superior spectral fidelity.

For the diffusion-based models, Diff-WaveNeXt 2 also produces synthesized speech quality comparable to that of FastDiff. By adopting the fixed noise conditioned sub-model training strategy proposed in [21], the speech quality is further improved. Notably, Diff-WaveNeXt 2 with sub-modeling achieves lower log F0 RMSE than HiFi-GAN. Compared to FastDiff, Diff-WaveNeXt 2 offers a 36% reduction in RTF on GPU and an 80% reduction on CPU, significantly enhancing inference efficiency.

All of the models' training times are detailed in Table 2. GAN-based models generally require substantial computational resources to train to convergence. For example, HiFi-GAN takes around 270 hours, while WaveFit and our proposed GAN-WaveNeXt 2 both require approximately 410 hours. In contrast, diffusion-based models are significantly more efficient. Our proposed Diff-WaveNeXt 2 requires only 32 hours of training, a major reduction compared to the 96 hours typically needed for FastDiff. This notable decrease in computational and time costs makes Diff-WaveNeXt 2 highly suitable for large-scale or resource-constrained applications, as it maintains acceptable speech synthesis quality with a much lighter training burden.

Although sub-modeling improves training efficiency and performance, it also increases the total size of the model, as described in Table 1. The overall parameters will grow with the number of sub-models. This is an issue of the proposed methods.

5. CONCLUSION

This paper introduces WaveNeXt 2, a unified ConvNeXt-based generator with a residual denoising sub-modeling structure that is the first to be compatible with both GAN- and diffusion-based neural vocoders. Our approach successfully addresses the performance limitations of the original GAN-based WaveNeXt in multi-speaker scenarios, where it previously underperformed compared to HiFi-GAN. By extending the use of ConvNeXt-based generators beyond the GAN framework, WaveNeXt 2 allows for a direct, intuitive comparison between the two major architectures. Our results demonstrate that both GAN-WaveNeXt 2 and Diff-WaveNeXt 2 achieve high performance. Specifically, GAN-WaveNeXt 2 provides significantly faster inference, especially on a CPU, while maintaining synthesis quality comparable to HiFi-GAN and WaveFit. Concurrently, Diff-WaveNeXt 2 delivers much faster CPU inference and superior perceptual quality compared to a 4-step FastDiff model, effectively outperforming it in both efficiency and quality. This unified framework offers flexible choices for various applications: for fast deployment in resource-constrained environments, Diff-WaveNeXt 2 is the ideal choice, whereas for scenarios demanding the highest synthesis quality, GAN-WaveNeXt 2 is the better option.

Acknowledgement: Part of this work was supported by JSPS KAKENHI Grant Numbers JP21H05054, JP23K21681, JP24K0296, and JP25H01139, as well as JST NEXUS (JPMJNX25C1).

6. REFERENCES

[1] K. Kumar, R. Kumar, T. de Boissiere, L. Gestin, W. Z. Teoh, J. Sotelo, A. de Brébisson, Y. Bengio, and A. C Courville, “MelGAN: Generative adversarial networks for conditional waveform synthesis,” in *Proc. NeurIPS*, Dec. 2019, pp. 14910–14921.

[2] G. Yang, S. Yang, K. Liu, P. Fang, W. Chen, and L. Xie, “Multi-band MelGAN: Faster waveform generation for high-quality text-to-speech,” in *Proc. SLT*, Jan. 2021, pp. 492–498.

[3] J. Kong, J. Kim, and J. Bae, “HiFi-GAN: Generative adversarial networks for efficient and high fidelity speech synthesis,” in *Proc. NeurIPS*, Dec. 2020, pp. 17022–17033.

[4] W. Jang, D. Lim, J. Yoon, B. Kim, and J. Kim, “UnivNet: A neural vocoder with multi-resolution spectrogram discriminators for high-fidelity waveform generation,” in *Proc. Interspeech*, Aug. 2021, pp. 2207–2211.

[5] T. Kaneko, K. Tanaka, H. Kameoka, and S. Seki, “iSTFTNet: Fast and lightweight mel-spectrogram vocoder incorporating inverse short-time Fourier transform,” in *Proc. ICASSP*, May 2022, pp. 6207–6211.

[6] S. Lee, W. Ping, B. Ginsburg, B. Catanzaro, and S. Yoon, “BigVGAN: A universal neural vocoder with large-scale training,” in *Proc. ICLR*, May 2023.

[7] H. Yamashita, T. Okamoto, R. Takashima, Y. Ohtani, T. Takiguchi, T. Toda, and H. Kawai, “Fast neural speech waveform generative models with fully-connected layer-based upsampling,” *IEEE Access*, vol. 12, pp. 31409–31421, 2024.

[8] H. Siuzdak, “Vocos: Closing the gap between time-domain and Fourier-based neural vocoders for high-quality audio synthesis,” in *Proc. ICLR*, May 2024.

[9] T. Okamoto, H. Yamashita, Y. Ohtani, T. Toda, and H. Kawai, “WaveNeXt: ConvNeXt-based fast neural vocoder without iSTFT layer,” in *Proc. ASRU*, Dec. 2023.

[10] T. Baoueb, H. Liu, M Fontaine, J. Le Roux, and G. Richard, “SpecDiff-GAN: A spectrally-shaped noise diffusion GAN for speech and music synthesis,” in *in Proc. ICASSP*, Apr. 2024, pp. 986–990.

[11] Y. Koizumi, K. Yatabe, H. Zen, and M. Bacchiani, “WaveFit: An iterative and non-autoregressive neural vocoder based on fixed-point iteration,” in *Proc. SLT*, Jan. 2023, pp. 884–891.

[12] Z. Kong, W. Ping, J. Huang, K. Zhao, and B. Catanzaro, “DiffWave: A versatile diffusion model for audio synthesis,” in *Proc. ICLR*, May 2021.

[13] N. Chen, Y. Zhang, H. Zen, R. J. Weiss, M. Norouzi, and W. Chan, “WaveGrad: Estimating gradients for waveform generation,” in *Proc. ICLR*, May 2021.

[14] S. Lee, H. Kim, C. Shin, X. Tan, C. Liu, Q. Meng, T. Qin, W. Chen, S. Yoon, and T.-Y. Liu, “PriorGrad: Improving conditional denoising diffusion models with data-dependent adaptive prior,” in *Proc. ICLR*, Apr. 2022.

[15] Y. Koizumi, H. Zen, K. Yatabe, N. Chen, and M. Bacchiani, “SpecGrad: Diffusion probabilistic model based neural vocoder with adaptive noise spectral shaping,” in *Proc. Interspeech*, Sept. 2022, pp. 803–807.

[16] R. Huang, M. W. Y. Lam, J. Wang, D. Su, D. Yu, Y. Ren, and Z. Zhao, “FastDiff: A fast conditional diffusion model for high-quality speech synthesis,” in *Proc. IJCAI*, July 2022, pp. 4157–4163.

[17] H. Tachibana, M. Inahara, M. Go, Y. Katayama, and Y. Watanabe, “Diffusion generative vocoder for fullband speech synthesis based on weak third-order SDE solver,” in *Proc. Interspeech*, Sept. 2022, pp. 1641–1645.

[18] N. Takahashi, M. Kumar, Singh, and Y. Mitsufuji, “Hierarchical diffusion models for singing voice neural vocoder,” in *Proc. ICASSP*, June 2023.

[19] R. Huang, Y. Ren, Z. J, C. Cui, J. Liu, and Z. Zhao, “FastDiff 2: Revisiting and incorporating GANs and diffusion models in high-fidelity speech synthesis,” in *Proc. ACL*, July 2023, pp. 6994–7009.

[20] T. D. Nguyen, J.-H. Kim, Y. Jang, J. Kim, and J. S. Chung, “FreGrad: Lightweight and fast frequency-aware diffusion vocoder,” in *Proc. ICASSP*, Apr. 2024, pp. 10736–10740.

[21] T. Okamoto, T. Toda, Y. Shiga, and H. Kawai, “Noise level limited sub-modeling for diffusion probabilistic vocoders,” in *Proc. ICASSP*, June 2021, pp. 6014–6018.

[22] M. W. Y. Lam, J. Wang, D. Su, and D. Yu, “BDDM: Bilateral denoising diffusion models for fast and high-quality speech synthesis,” in *Proc. ICLR*, Apr. 2022.

[23] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, “A ConvNet for the 2020s,” in *Proc. CVPR*, June 2022, pp. 11976–11986.

[24] T. Okamoto, Y. Ohtani, T. Toda, and H. Kawai, “ConvNeXt-TTS and ConvNeXt-VC: ConvNeXt-based fast end-to-end sequence-to-sequence text-to-speech and voice conversion,” in *Proc. ICASSP*, Apr. 2024, pp. 12456–12460.

[25] P. L. Combettes and J.-C. Pesquet, “Fixed point strategies in data science,” *IEEE Trans. Signal Process.*, vol. 69, pp. 3878–3905, 2021.

[26] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, “PyTorch: An imperative style, high-performance deep learning library,” in *Proc. NeurIPS*, Dec. 2019, pp. 8024–8035.

[27] Y. Koizumi, H. Zen, S. Karita, Y. Ding, K. Yatabe, N. Morioka, M. Bacchiani, Y. Zhang, W. Han, and A. Bapna, “LibriTTS-R: A restored multi-speaker text-to-speech corpus,” in *Proc. Interspeech*, Aug. 2023, pp. 5496–5500.

[28] ITU-T Recommendation P. 800, *Methods for subjective determination of transmission quality*, 1996.

[29] T. Saeki, D. Xin, W. Nakata, T. Koriyama, S. Takamichi, and H. Saruwatari, “UTMOS: UTokyo-SaruLab system for VoiceMOS Challenge 2022,” in *Proc. Interspeech*, Sept. 2022, pp. 4521–4525.

[30] G. Mittag, B. Naderi, A. Chehadi, and S. Möller, “NISQA: A deep CNN-self-attention model for multidimensional speech quality prediction with crowdsourced datasets,” in *Proc. Interspeech*, Aug. 2021, pp. 2127–2131.

[31] R. Kubichek, “Mel-cepstral distance measure for objective speech quality assessment,” in *Proc. PACRIM*, May 1993, vol. 1, pp. 125–128.