ISTFTNET: FAST AND LIGHTWEIGHT MEL-SPECTROGRAM VOCODER INCORPORATING INVERSE SHORT-TIME FOURIER TRANSFORM

Takuhiro Kaneko, Kou Tanaka, Hirokazu Kameoka, Shogo Seki

NTT Communication Science Laboratories, NTT Corporation, Japan

ABSTRACT

In recent text-to-speech synthesis and voice conversion systems, a mel-spectrogram is commonly applied as an intermediate representation, and the necessity for a mel-spectrogram vocoder is increasing. A mel-spectrogram vocoder must solve three inverse problems: recovery of the original-scale magnitude spectrogram, phase reconstruction, and frequency-to-time conversion. A typical convolutional mel-spectrogram vocoder solves these problems jointly and implicitly using a convolutional neural network, including temporal upsampling layers, when directly calculating a raw waveform. Such an approach allows skipping redundant processes during waveform synthesis (e.g., the direct reconstruction of high-dimensional original-scale spectrograms). By contrast, the approach solves all problems in a black box and cannot effectively employ the time-frequency structures existing in a mel-spectrogram. We thus propose *iSTFTNet*, which replaces some output-side layers of the mel-spectrogram vocoder with the inverse short-time Fourier transform (iSTFT) after sufficiently reducing the frequency dimension using upsampling layers, reducing the computational cost from black-box modeling and avoiding redundant estimations of high-dimensional spectrograms. During our experiments, we applied our ideas to three HiFi-GAN variants and made the models faster and more lightweight with a reasonable speech quality.[1]

Index Terms— Waveform synthesis, mel-spectrogram vocoder, convolutional neural network, inverse short-time Fourier transform, generative adversarial networks

1. INTRODUCTION

Speech is a frequently used modality in communication, and text-to-speech (TTS) synthesis and voice conversion (VC) have been studied to eliminate human-human and human-machine boundaries. In both TTS and VC, typical methods use a two-stage approach: (1) The first model predicts the target intermediate representation from the text or source intermediate representations. (2) The second step generates a raw waveform from the predicted intermediate representation. A mel-spectrogram is widely used as an intermediate representation in recent TTS [1, 2, 3, 4, 5] and VC [6, 7, 8] systems owing to its compactness and expressiveness. Consequently, the demand for a mel-spectrogram vocoder is increasing.
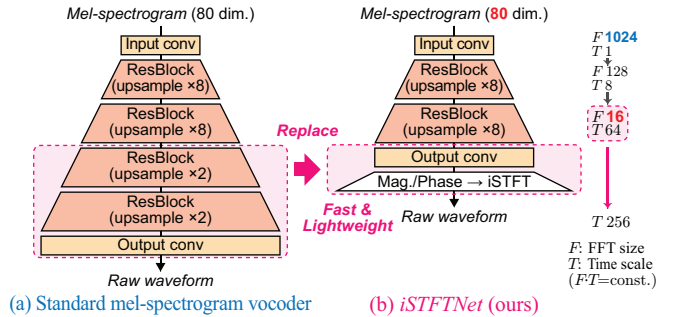
Fig. 1. Comparison of a standard convolutional mel-spectrogram vocoder and *iSTFTNet* (ours). We propose replacing the output-side layers of the standard vocoder (a) with iSTFT (b) when the number of frequency dimensions is sufficiently small (e.g., herein, the FFT size is 16) compared to the number of dimensions of the input mel-spectrogram (80).

A mel-spectrogram vocoder must solve the following three inverse problems: recovery of the original-scale magnitude spectrogram, phase reconstruction, and frequency-to-time conversion. A typical convolutional mel-spectrogram vocoder (e.g., a generative adversarial network (GAN [9]) based model [10, 11, 12]) solves these problems jointly and implicitly using a convolutional neural network (CNN), including temporal upsampling layers, when directly calculating a raw waveform from a mel-spectrogram. Such an approach allows omitting redundant processes during waveform synthesis, e.g., the direct reconstruction of high-dimensional original-scale spectrograms. However, this approach solves all problems in a black box and cannot efficiently employ time-frequency structures that exist in a mel-spectrogram.

We thus propose *iSTFTNet*, which replaces some output-side layers of the convolutional mel-spectrogram vocoder (Fig. 1(a)) with well-established signal processing, particularly an inverse short-time Fourier transform (iSTFT) (Fig. 1(b)) when the number of frequency dimensions (FFT size of 16, Fig. 1) is sufficiently small compared to the number of dimensions of the input mel-spectrogram (80, Fig. 1). This reduces the computational cost from black-box modeling while avoiding redundant estimations of high-dimensional original-scale spectrograms (FFT size of 1024, Fig. 1). During our experiments, we applied our ideas to three HiFi-GAN variants [12] and made the models faster and more lightweight with a reasonable speech quality.

The rest of this paper is organized as follows. In Section 2, we discuss related studies. In Section 3, we review a typical

---

[1] Audio samples are available at `https://www.kecl.ntt.co.jp/people/kaneko.takuhiro/projects/istftnet/`.

convolutional mel-spectrogram vocoder and introduce *iSTFT-Net*, which is a fast and lightweight variant. In Section 4, we present the experiment results. In Section 5, we provide some concluding remarks and areas of future research.

2. RELATED WORK

Neural vocoders have been studied in speech signal processing and machine learning. The first breakthrough was achieved using autoregressive models, including WaveNet [13] and WaveRNN [14], which achieved an impressive quality but slow inference speed owing to a sample-by-sample estimation. Parallelizable non-autoregressive models have therefore gained attention. For example, Parallel WaveNet [15] and ClariNet [16] distill an autoregressive teacher model into a non-autoregressive convolutional student model. WaveGlow [17] eliminates the requirement for a teacher model by incorporating Glow [18], composed of affine coupling layers and a 1×1 invertible convolution. WaveGrad [19] and DiffWave [20], based on diffusion probabilistic models [21, 22], apply non-autoregressive CNNs for parallel computations. A GAN [9]-based model [10, 11, 12, 23, 24, 25] achieves parallelizable training and inference through noncausal convolutions. As described, CNNs with temporal upsampling layers, shown in Fig. 1(a), have been commonly used in recent mel-spectrogram vocoders. Thus, beyond the HiFi-GANs [12] used in our experiments, our ideas are general and can be applied to other models.

The use of iSTFT for neural speech synthesis was previously introduced [26, 27, 28] (including our own early attempts [26]). As the main difference between the previous models and *iSTFTNet*, the former requires a high-capacity or high-computational model (e.g., 12 residual blocks with 2048 channels [28] and 2D CNNs [26, 27]) because they aim to reconstruct the original-scale spectrograms without changing the time scale. By contrast, *iSTFTNet* employs a hybrid approach in which iSTFT is applied after some upsampling processes (Fig. 1(b)). This allows a reasonable performance using a low-capacity model (e.g., 1D CNNs, commonly used in typical GAN vocoders [10, 11, 12, 23, 24, 25]).

3. METHOD

3.1. Convolutional mel-spectrogram vocoder

As shown in Fig. 2, the mel-spectrogram is extracted from the raw waveform as follows: (1) The magnitude and phase spectrograms are extracted from the raw waveform using a short-time Fourier transform (STFT). (2) The phase spectrogram is dropped. (3) The magnitude spectrogram is converted into a mel-scale. Because a mel-spectrogram vocoder is aimed at an inverse process, three inverse problems must be solved: (3') recovery of the original-scale magnitude spectrogram; (2') phase reconstruction; and (1') frequency-to-time conversion.

A typical convolutional mel-spectrogram vocoder solves these problems jointly and implicitly using a CNN, including temporal upsampling layers, while directly calculating a raw waveform from a mel-spectrogram. This approach al-
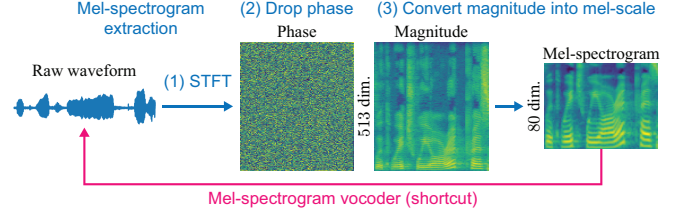
Fig. 2. Processing flows of mel-spectrogram extraction (light blue) and mel-spectrogram vocoder (pink)

lows redundant processes (e.g., the direct reconstruction of high-dimensional original-scale magnitude and phase spectrograms) to be skipped during waveform synthesis. This allows a convolutional mel-spectrogram vocoder to solve the aforementioned problems with a low-capacity model. For example, HiFi-GAN V2 [12] achieves a good performance using only 1D convolutions of channels smaller than 128, despite being smaller than the original-scale spectrogram dimensions (i.e., 513, Fig. 2).

3.2. iSTFTNet: Fast and lightweight vocoder with iSTFT

Black-box modeling is useful for discovering potential shortcuts. However, we cannot effectively employ the time-frequency structures existing in the mel-spectrogram despite providing a hint for solving inverse problems.

Thus, we propose *iSTFTNet*, which employs time and frequency structures explicitly using iSTFT after sufficiently reducing the frequency dimension using some upsampling layers, as shown in Fig. 3(b)–(d). Here, we utilize the characteristics of STFT, that is, the trade-off between the time and frequency resolution. More precisely, when the iSTFT required after $s\times$ upsampling is represented as $\mathrm{iSTFT}(f_s, h_s, w_s)$, where f_s, h_s, and w_s indicate the FFT size, hop length, and window length, respectively; $\mathrm{iSTFT}(f_s, h_s, w_s)$ can be calculated using the parameters of iSTFT required for the original-scale spectrogram, $\mathrm{iSTFT}(f_1, h_1, w_1)$:

$$\mathrm{iSTFT}(f_s, h_s, w_s) = \mathrm{iSTFT}\left(\frac{f_1}{s}, \frac{h_1}{s}, \frac{w_1}{s}\right), \qquad (1)$$

where we utilize the aforementioned STFT characteristic, that is, $f_1 \cdot 1 = f_s \cdot s = \text{constant}$. This equation means that we can reduce the frequency dimensions by increasing s. As shown in Fig. 3, we can simplify the structure in the frequency direction by increasing the number of upsamples. In Section 4.2, we empirically found that simplification through more than two upsamples (Fig. 3(c) or (d)) is essential for a faster and more lightweight model to achieve a reasonable quality.

3.3. Implementation

iSTFTNets (Fig. 3(b)–(d)) have the mostly same network architecture as the baseline (Fig. 3(e)). Hence, when a reliable convolutional mel-spectrogram vocoder is obtained, it is easy to incorporate the concept of *iSTFTNet*. However, three minor but essential modifications are required: (i) The output channels of the final convolutional layer should be changed from 1 to $(f_s/2 + 1) \times 2$ to generate magnitude and phase
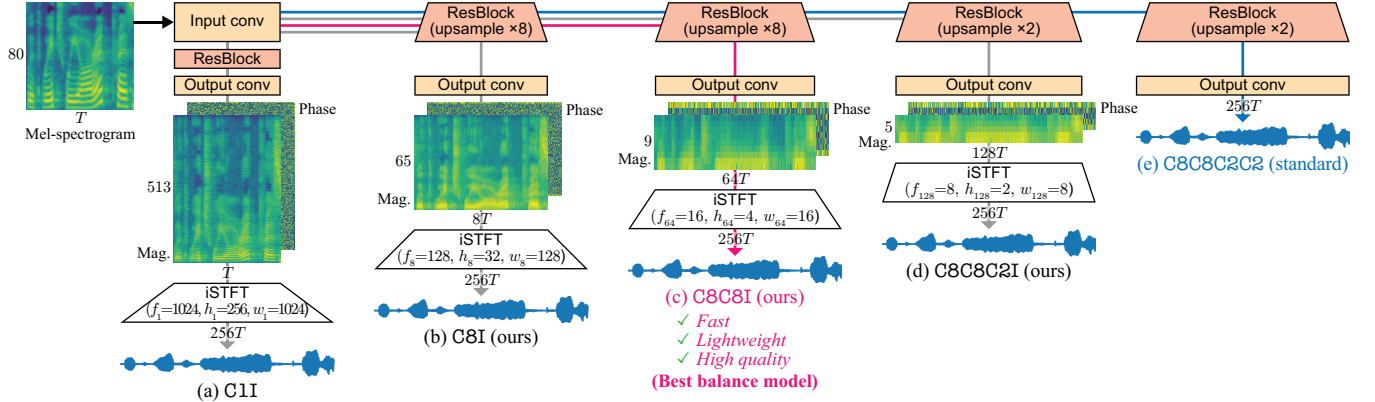
Fig. 3. Architectures of *iSTFTNets* (b)–(d) and a standard convolutional mel-spectrogram vocoder (e). The model is denoted as `Cx...(I)`, where `Cx` indicates the use of a residual block (ResBlock) [29] with an $\times x$ upsampling layer and `I` indicates the use of iSTFT. Here, the input 80-dimensional mel-spectrogram was extracted from a 22.05-kHz waveform using STFT with an FFT size of 1024, hop length of 256, and window length of 1024.

spectrograms instead of a raw waveform. (ii) Exponential and sine activation functions should be applied to the output of (i) when calculating the magnitude and phase spectrograms, respectively. (iii) A raw waveform should be generated from the magnitude and phase spectrograms using iSTFT (Eq. (1)). For (ii), we use an exponential activation function because the required magnitude spectrogram uses a linear scale, whereas the input mel-spectrogram uses a log scale, and we apply a sine activation function to represent the periodic characteristics of the phase spectrogram.

4. EXPERIMENTS

4.1. Experiment setup

Dataset. We used the LJSpeech dataset [30], consisting of 13,100 audio clips (24 h) of a female speaker. Here, 12,600, 250, and 250 utterances were used for the training, validation, and evaluation, respectively. The audio clips were sampled at 22.05 kHz, and 80-dimensional log-mel spectrograms were extracted with an FFT size of 1024, hop length of 256, and window length of 1024.

Network architectures. We applied our ideas to three HiFi-GAN variants [12] (high-quality (V1), light (V2), and carefully tuned (V3) variants). We implemented them based on an open-source code[2] for fair comparison with the various synthesis speeches provided. As mentioned in Section 3.3, with the exception of the three modifications described above, we used the same architectures as the baselines.

Training settings. We trained the models using the HiFi-GAN configuration provided in the open-source code[2], the parameters of which were tuned for stable training across various datasets. We trained the model for 2.5M iterations using the Adam optimizer [31] with an initial learning rate of 0.0002, and momentum terms β_1 and β_2 of 0.5 and 0.9, respectively. For the loss function, we used a combination of least squares GAN [32], mel-spectrogram [12], and feature matching [33, 10] losses.

---

[2]https://github.com/kan-bayashi/ParallelWaveGAN

4.2. Evaluation

We conducted a mean opinion score (MOS) test to evaluate the perceptual quality, randomly selecting 20 utterances from the evaluation set and using the ground truth mel-spectrograms of the utterances as the vocoder input. This test was conducted online with 16 listeners. Audio samples are available from the link[1] presented on the first page. As an objective metric, we used the conditional Fréchet wav2vec distance (*cFW2VD*), which measures the distance between real and generative distributions in a wav2vec 2.0 [34] feature space conditioned on text information. This is conceptually similar to the Fréchet inception distance (FID) [35] and Fréchet DeepSpeech distance (FDSD) [36], which measure the perceptual quality of images and speeches, respectively. We used cFW2VD instead of conditional FDSD (cFDSD) [36] to evaluate the raw waveform directly without converting it into a power spectrogram, as required in cFDSD. We found that MOS has a higher correlation with cFW2VD than with cFDSD (Spearman's rank correlation of -0.93 and -0.83, respectively). In cFW2VD, the smaller the value is, the better the perceptual quality. Table 1 shows the results, along with the inference speed and model size. We examined the approach from three perspectives.

(1) How many layers should be replaced with iSTFT? The vocoder is improved by replacing its output-side layers with a faster and more lightweight iSTFT. To determine the number of layers to be replaced, we investigated the performance differences between the models shown in Fig. 3. The corresponding results are listed in Table 1 (Nos. 2–5, 7–10, and 12–14).[3] As expected, the inference speeds up, and the model size decreases with more replaced layers. For the MOS, we found that `C8I` performs worse than the original in all cases; however, `C8C8I` and `C8C8C2I` for V1 and V2 were comparable to the original. For V3 only, the performance decreases when `C8C8I` is used. This is because V3 is carefully tuned to reduce the number of layers and loses its generality. How-

---

[3]`C8C8C2I` was not used for V3 because the network of V3 is `C8C8C4` and does not have a fourth upsampling layer, differently from V1 and V2.

Table 1. Comparison of MOS with 95% confidence intervals, cFW2VD, inference speed, and model size. The inference speed (relative speed compared to real time) using a GPU was calculated on a single NVIDIA V100 GPU, and the speed using a CPU was computed on a MacBook Pro laptop (2.7-GHz Intel Core i7). We report the average score over the utterances in the evaluation set. The model identifier (e.g., `C8C8I`) is shown in Fig. 3. The numbers in () indicate the rates (%) compared with the baselines (V1, V2, or V3). The underlined models are *iSTFTNets* (fast and lightweight models).

No.	Model	MOS↑	cFW2VD↓	Speed↑ (GPU)	Speed↑ (CPU)	# Param↓ (M)
1	Ground truth	4.46 ±0.14	–	–	–	–
2	V1 (original) [12]	4.22 ±0.17	0.020	×143.59 (100)	×1.34 (100)	13.94 (100)
3	V1-C8C8C2I	4.22 ±0.17	0.018	×179.42 (125)	×1.63 (122)	13.80 (99)
4	V1-C8C8I	4.26 ±0.17	0.020	×245.68 (171)	×2.33 (174)	13.26 (95)
5	V1-C8I	3.32 ±0.22	0.073	×609.43 (424)	×7.57 (565)	10.89 (78)
6	V1-C8C1I	3.82 ±0.17	0.033	×326.39 (227)	×3.97 (296)	19.15 (137)
7	V2 (original) [12]	3.91 ±0.17	0.046	×624.47 (100)	×10.39 (100)	0.93 (100)
8	V2-C8C8C2I	3.98 ±0.17	0.038	×732.96 (117)	×13.34 (128)	0.92 (99)
9	V2-C8C8I	3.95 ±0.16	0.042	×1025.46 (164)	×20.37 (196)	0.89 (96)
10	V2-C8I	3.21 ±0.20	0.096	×1720.91 (276)	×68.05 (655)	0.78 (84)
11	V2-C8C1I	3.44 ±0.20	0.071	×1081.37 (173)	×39.14 (377)	1.30 (140)
12	V3 (original) [12]	3.78 ±0.16	0.052	×933.06 (100)	×10.40 (100)	1.46 (100)
13	V3-C8C8I	3.41 ±0.19	0.055	×1517.70 (163)	×21.48 (206)	1.42 (97)
14	V3-C8I	2.89 ±0.17	0.156	×2481.87 (266)	×66.83 (642)	1.28 (87)
15	V3-C8C1I	2.82 ±0.21	0.116	×1925.15 (206)	×41.16 (396)	1.77 (121)
16	MB-MelGAN [24]	3.54 ±0.21	0.078	×1070.95	×17.95	2.54
17	PWG [11]	3.47 ±0.21	0.066	×79.71	×0.70	1.35

ever, note that despite the performance decrease, V3-`C8C8I` is still comparable with Parallel WaveGAN (PWG) [11] (Table 1 (No. 17)), while improving the inference speed.

(2) Necessity of combining upsampling and iSTFT. The numbers of both the upsampling layers and residual blocks differ between `C8I` and `C8C8I`, as shown in Fig. 3(b) and (c). To solve this problem, we examined the performance of `C8C1I`[4] which applies one upsampling but uses two residual blocks, similar to `C8C8I`. The corresponding results are presented in Table 1 (Nos. 6, 11, and 15). We found that `C8C1I` is still worse than `C8C8I`, indicating that reducing the frequency dimension using upsampling is essential for obtaining a reasonable quality when applying iSTFT without significant changes to the network architecture.[5]

(3) Comparison with fastest baseline. One of the fastest GAN vocoders is multi-band (MB) MelGAN [24], which increases the speed of MelGAN [10] by changing the synthesis target from a full-band signal to lower-resolution sub-band signals [37]. To examine the validity of this speed, we compared our models with MB-MelGAN. The corresponding results are presented in Table 1 (No. 16). Here, V2-`C8C8I` outperformed MB-MelGAN for MOS, reducing the model size and achieving a comparable speed. Note that the multi-band formulation and iSTFT are orthogonal and compatible, and iSTFT can be incorporated into MB-MelGAN using

---

[4]The model size of `C8C1I` is larger than that of `C8C8I` because in the latter, the channels are halved in the second ResBlock after upsampling, whereas in the former, this is not conducted owing to the absence of upsampling.

[5]We also examined non-upsampling models (particularly, `C1I` (Fig. 3(a)) and `C1C1I`). Finding that they suffer from training difficulties with a significantly lower speech quality, we omitted them from the experiments.

Table 2. Comparison of MOS with 95% confidence intervals and cFW2VD for TTS synthesis

No.	Model	MOS↑	cFW2VD↓
1	Ground truth	4.32 ±0.10	–
2	Conformer-FS2 + V1	4.09 ±0.12	0.216
3	Conformer-FS2 + V1-C8C8I	4.25 ±0.11	0.214
4	Conformer-FS2 [38]	3.66 ±0.15	0.242

iSTFT $\left(\frac{f_1}{sb}, \frac{h_1}{sb}, \frac{w_1}{sb}\right)$, where b is the number of sub-bands. This approach remains for future research.[6]

4.3. Application to text-to-speech synthesis

Next, we examine the effectiveness of our approach when applied to TTS synthesis, focusing on the difference in performance between V1 and V1-`C8C8I` when combined with Conformer-FS2 [38] (a combination of Conformer [39] and FastSpeech 2 (FS2) [5]). Following [12], which shows the utility of fine-tuning on HiFi-GAN, we fine-tuned the combined models for 300k iterations in an end-to-end manner after training each model. We applied an open-source code [40][7] for fair comparison with other speech samples provided. We conducted a MOS test to evaluate the perceptual quality by randomly selecting 20 utterances from the evaluation set. This test was conducted online with 12 listeners. Audio samples are available from the link[1] presented on the first page.

Table 2 summarizes the results. We found that V1-`C8C8I` not only achieves a comparable or better performance than V1 and Conformer-FS2, but also is comparable with the ground truth. These results indicate that *iSTFTNet* does not compromise the speech quality, even for TTS synthesis.

5. CONCLUSION

To employ time-frequency structures in a mel-spectrogram while avoiding redundant estimations of high-dimensional original-scale spectrograms, we propose *iSTFTNet*, replacing the output-side layers of a convolutional mel-spectrogram vocoder with iSTFT after reducing the frequency dimension using upsampling layers. The experiment results demonstrate that we can make the models faster and more lightweight using iSTFT, and that upsampling processes are essential for obtaining a reasonable quality. As discussed in Section 2, CNNs with upsampling layers are used by various vocoders beyond GAN-based versions. Hence, applying our idea to such vocoders will be interesting. We also concurrently investigate the utility of the inverse fast Fourier transform for a recurrent neural vocoder, and plan to examine the difference in performance in future studies to further validate the utility of the inverse Fourier transform.

---

[6]As another difference between *iSTFTNet* and MB-MelGAN, MB-MelGAN requires an additional sub-band STFT loss for stable training, whereas *iSTFTNet* does not.

[7]https://github.com/espnet/espnet

6. REFERENCES

[1] 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, pp. 4779–4783.

[2] Wei Ping, Kainan Peng, Andrew Gibiansky, Sercan O. Arik, Ajay Kannan, Sharan Narang, Jonathan Raiman, and John Miller, "Deep Voice 3: Scaling text-to-speech with convolutional sequence learning," in *Proc. ICLR*, 2018.

[3] Naihan Li, Shujie Liu, Yanqing Liu, Sheng Zhao, and Ming Liu, "Neural speech synthesis with transformer network," in *Proc. AAAI*, 2019, pp. 6706–6713.

[4] Yi Ren, Yangjun Ruan, Xu Tan, Tao Qin, Sheng Zhao, Zhou Zhao, and Tie-Yan Liu, "FastSpeech: Fast, robust and controllable text to speech," in *Proc. NeurIPS*, 2019.

[5] Yi Ren, Chenxu Hu, Xu Tan, Tao Qin, Sheng Zhao, Zhou Zhao, and Tie-Yan Liu, "FastSpeech 2: Fast and high-quality end-to-end text to speech," in *Proc. ICLR*, 2021.

[6] Kaizhi Qian, Yang Zhang, Shiyu Chang, Xuesong Yang, and Mark Hasegawa-Johnson, "Auto-VC: Zero-shot voice style transfer with only autoencoder loss," in *Proc. ICML*, 2019, pp. 5210–5219.

[7] Takuhiro Kaneko, Hirokazu Kameoka, Kou Tanaka, and Nobukatsu Hojo, "CycleGAN-VC3: Examining and improving CycleGAN-VCs for mel-spectrogram conversion," in *Proc. Interspeech*, 2020, pp. 2017–2021.

[8] Takuhiro Kaneko, Hirokazu Kameoka, Kou Tanaka, and Nobukatsu Hojo, "MaskCycleGAN-VC: Learning non-parallel voice conversion with filling in frames," in *Proc. ICASSP*, 2021, pp. 5919–5923.

[9] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio, "Generative adversarial nets," in *Proc. NIPS*, 2014, pp. 2672–2680.

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

[11] Ryuichi Yamamoto, Eunwoo Song, and Jae-Min Kim, "Parallel Wave-GAN: A fast waveform generation model based on generative adversarial networks with multi-resolution spectrogram," in *Proc. ICASSP*, 2020, pp. 6199–6203.

[12] Jungil Kong, Jaehyeon Kim, and Jaekyoung Bae, "HiFi-GAN: Generative adversarial networks for efficient and high fidelity speech synthesis," in *Proc. NeurIPS*, 2020.

[13] Aäron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu, "WaveNet: A generative model for raw audio," *arXiv preprint arXiv:1609.03499*, 2016.

[14] Nal Kalchbrenner, Erich Elsen, Karen Simonyan, Seb Noury, Norman Casagrande, Edward Lockhart, Florian Stimberg, Aäron van den Oord, Sander Dieleman, and Koray Kavukcuoglu, "Efficient neural audio synthesis," in *Proc. ICML*, 2018, pp. 2410–2419.

[15] Aäron van den Oord, Yazhe Li, Igor Babuschkin, Karen Simonyan, Oriol Vinyals, Koray Kavukcuoglu, George van den Driessche, Edward Lockhart, Luis Cobo, Florian Stimberg, Norman Casagrande, Dominik Grewe, Seb Noury, Sander Dieleman, Erich Elsen, Nal Kalchbrenner, Heiga Zen, Alex Graves, Helen King, Tom Walters, Dan Belov, and Demis Hassabis, "Parallel WaveNet: Fast high-fidelity speech synthesis," in *Proc. ICML*, 2018, pp. 3918–3926.

[16] Wei Ping, Kainan Peng, and Jitong Chen, "ClariNet: Parallel wave generation in end-to-end text-to-speech," in *Proc. ICLR*, 2019.

[17] Ryan Prenger, Rafael Valle, and Bryan Catanzaro, "WaveGlow: A flow-based generative network for speech synthesis," in *Proc. ICASSP*, 2019, pp. 3617–3621.

[18] Diederik P. Kingma and Prafulla Dhariwal, "Glow: Generative flow with invertible 1×1 convolutions," in *Proc. NeurIPS*, 2018.

[19] Nanxin Chen, Yu Zhang, Heiga Zen, Ron J. Weiss, Mohammad Norouzi, and William Chan, "WaveGrad: Estimating gradients for waveform generation," in *Proc. ICLR*, 2020.

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

[21] Yang Song and Stefano Ermon, "Generative modeling by estimating gradients of the data distribution," in *Proc. NeurIPS*, 2019.

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

[23] Jinhyeok Yang, Junmo Lee, Youngik Kim, Hoonyoung Cho, and Injung Kim, "VocGAN: A high-fidelity real-time vocoder with a hierarchically-nested adversarial network," in *Proc. Interspeech*, 2020, pp. 200–204.

[24] Geng Yang, Shan Yang, Kai Liu, Peng Fang, Wei Chen, and Lei Xie, "Multi-band MelGAN: Faster waveform generation for high-quality text-to-speech," in *Proc. SLT*, 2021, pp. 492–498.

[25] Ahmed Mustafa, Nicola Pia, and Guillaume Fuchs, "StyleMelGAN: An efficient high-fidelity adversarial vocoder with temporal adaptive normalization," in *Proc. ICASSP*, 2021, pp. 6034–6038.

[26] Keisuke Oyamada, Hirokazu Kameoka, Takuhiro Kaneko, Kou Tanaka, Nobukatsu Hojo, and Hiroyasu Ando, "Generative adversarial network-based approach to signal reconstruction from magnitude spectrogram," in *Proc. EUSIPCO*, 2018, pp. 2514–2518.

[27] Paarth Neekhara, Chris Donahue, Miller Puckette, Shlomo Dubnov, and Julian McAuley, "Expediting TTS synthesis with adversarial vocoding," in *Proc. Interspeech*, 2019, pp. 186–190.

[28] Alexey A. Gritsenko, Tim Salimans, Rianne van den Berg, Jasper Snoek, and Nal Kalchbrenner, "A spectral energy distance for parallel speech synthesis," in *Proc. NeurIPS*, 2020.

[29] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *Proc. CVPR*, 2016, pp. 770–778.

[30] Keith Ito and Linda Johnson, "The LJ speech dataset," https://keithito.com/LJ-Speech-Dataset/, 2017.

[31] Diederik P. Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," in *Proc. ICLR*, 2015.

[32] Xudong Mao, Qing Li, Haoran Xie, Raymond Y.K. Lau, Zhen Wang, and Stephen Paul Smolley, "Least squares generative adversarial networks," in *Proc. ICCV*, 2017, pp. 2794–2802.

[33] Anders Boesen Lindbo Larsen, Søren Kaae Sønderby, Hugo Larochelle, and Ole Winther, "Autoencoding beyond pixels using a learned similarity metric," in *Proc. ICML*, 2016, pp. 1558–1566.

[34] Alexei Baevski, Henry Zhou, Abdelrahman Mohamed, and Michael Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," in *Proc. NeurIPS*, 2020.

[35] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter, "GANs trained by a two time-scale update rule converge to a local Nash equilibrium," in *Proc. NIPS*, 2017.

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

[37] Chengzhu Yu, Heng Lu, Na Hu, Meng Yu, Chao Weng, Kun Xu, Peng Liu, Deyi Tuo, Shiyin Kang, Guangzhi Lei, Dan Su, and Dong Yu, "DurIAN: Duration informed attention network for multimodal synthesis," in *Proc. Interspeech*, 2020, pp. 2027–2031.

[38] Pengcheng Guo, Florian Boyer, Xuankai Chang, Tomoki Hayashi, Yosuke Higuchi, Hirofumi Inaguma, Naoyuki Kamo, Chenda Li, Daniel Garcia-Romero, Jiatong Shi, Jing Shi, Shinji Watanabe, Kun Wei, Wangyou Zhang, and Yuekai Zhang, "Recent developments on ESPnet toolkit boosted by conformer," in *Proc. ICASSP*, 2021, pp. 5874–5878.

[39] Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, and Ruoming Pang, "Conformer: Convolution-augmented transformer for speech recognition," in *Proc. Interspeech*, 2020, pp. 5036–5040.

[40] Tomoki Hayashi, Ryuichi Yamamoto, Katsuki Inoue, Takenori Yoshimura, Shinji Watanabe, Tomoki Toda, Kazuya Takeda, Yu Zhang, and Xu Tan, "ESPnet-TTS: Unified, reproducible, and integratable open source end-to-end text-to-speech toolkit," in *Proc. ICASSP*, 2020, pp. 7654–7658.