Interspeech 2022
18-22 September 2022, Incheon, Korea

MISRNet: Lightweight Neural Vocoder Using Multi-Input Single Shared Residual Blocks

Takuhiro Kaneko, Hirokazu Kameoka, Kou Tanaka, Shogo Seki

NTT Communication Science Laboratories, NTT Corporation, Japan

{takuhiro.kaneko.tb, hirokazu.kameoka.uh, kou.tanaka.ef, shogo.seki.va}@hco.ntt.co.jp

Abstract

Neural vocoders have recently become popular in text-to-speech synthesis and voice conversion, increasing the demand for efficient neural vocoders. One successful approach is HiFi-GAN, which archives high-fidelity audio synthesis using a relatively small model. This characteristic is obtained using a generator incorporating multi-receptive field fusion (MRF) with multiple branches of residual blocks, allowing the expansion of the description capacity with few-channel convolutions. However, MRF requires the model size to increase with the number of branches. Alternatively, we propose a network called *MISRNet*, which incorporates a novel module called *multi-input single shared residual block (MISR)*. MISR enlarges the description capacity by enriching the input variation using lightweight convolutions with a kernel size of 1 and, alternatively, reduces the variation of residual blocks from multiple to single. Because the model size of the input convolutions is significantly smaller than that of the residual blocks, MISR reduces the model size compared with that of MRF. Furthermore, we introduce an implementation technique for MISR, where we accelerate the processing speed by adopting tensor reshaping. We experimentally applied our ideas to lightweight variants of HiFi-GAN and iSTFTNet, making the models more lightweight with comparable speech quality and without compromising speed.[1]

Index Terms: waveform synthesis, neural vocoder, lightweight, weight sharing, generative adversarial networks

1. Introduction

Speech is essential for human-human and human-machine interactions. Text-to-speech (TTS) synthesis and voice conversion (VC) have been investigated to eliminate these boundaries. In typical TTS [1, 2, 3, 4] and VC [5, 6, 7, 8] systems, a two-stage approach is used. (1) The first model predicts the target intermediate representation (e.g., mel spectrogram) from the text or source intermediate representation. (2) The second model synthesizes a waveform from a predicted intermediate representation. Neural vocoders handle the second step. To widen their applications, there is an increasing demand for efficient neural vocoders that can be incorporated into various devices and conditions.

The first breakthrough in neural vocoder studies was brought about by autoregressive models (e.g., WaveNet [9] and WaveRNN [10]), which achieve high-quality audio synthesis but slow inference because of their frame-by-frame estimation. To improve inference speed, parallelizable non-autoregressive models have been considered. For instance, Parallel WaveNet [11] and ClariNet [12] distill an autoregressive teacher model into a non-autoregressive student model. To exclude the requirement for a teacher model, WaveGlow [13] incorporates Glow [14], which includes affine coupling layers and 1×1 invertible convolutions. WaveGrad [15] and DiffWave [16] adopt diffusion probabilistic models [17, 18] in which a waveform is created from white noise iteratively using a gradient-based sampler.

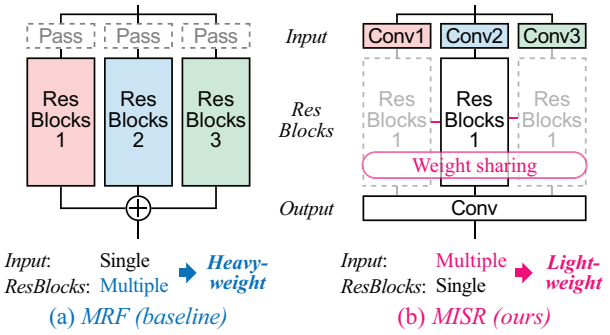

Figure 1: *Comparison between multi-receptive field fusion (MRF) [22] and the proposed multi-input single shared residual block (MISR).*

Another common approach is the generative adversarial network (GAN [19])-based model, which can train a non-autoregressive model without a teacher model. This approach has architectural flexibility and various variants (e.g., [20, 21, 22, 23, 24, 25, 26]) have been proposed. One successful approach is HiFi-GAN [22], which achieves high-fidelity speech synthesis using a relatively small model. Specifically, HiFi-GAN V2 (a lightweight variant) with approximately 0.9M parameters has better speech quality than MelGAN [20] with 4.3M parameters and WaveNet [9, 11] with 24.7M parameters. This advantage can be attributed to the incorporation of multi-receptive field fusion (MRF) into the generator. As shown in Figure 1(a), MRF has multiple branches of residual blocks [27], which allows the expansion of the description capacity with few-channel convolutions. However, in exchange for an increase in descriptive power, MRF requires the model size to increase with the number of branches.

As an alternative, we propose a network called *MISRNet*, which incorporates a novel module called *multi-input single shared residual block (MISR)*. As shown in Figure 1(b), MISR increases the description capacity by enriching the input variation using lightweight convolutions with a kernel size of 1 and, alternatively, reduces the variation of residual blocks from multiple to single. Because the model size of the input convolutions is significantly smaller than that of the residual blocks, MISR reduces the model size compared with that of MRF.

When implementing MISR naively, its processing speed is slower than that of MRF because input and output convolutions are additionally used, whereas the number of branches of the

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

10.21437/Interspeech.2022-11152

residual blocks applied in the process remains the same. To improve inference speed, we apply tensor reshaping before applying residual blocks and arrange each branch of the residual blocks in a batch dimension. Then, we can simultaneously adopt residual blocks with a typical fast batchwise operation.

We experimentally applied our concepts to lightweight variants of HiFi-GAN [22] and iSTFTNet [26], making them lighter with comparable speech quality and without compromising speed. We also show that *MISRNets* achieve better speech quality than networks with other lightweight modules (i.e., depthwise separable convolution (DSC) [28, 29, 30, 31]).

The rest of this paper is organized as follows. In Section 2, we first review the previous MRF and introduce the proposed MISR. Subsequently, we present an implementation technique for accelerating MISR. In Section 3, we present the experimental results. In Section 4, we present our concluding remarks and areas for future research.

2. Method

2.1. Previous: Multi-receptive field fusion

First, we explain MRF [22], a state-of-the-art efficient module used in HiFi-GAN. MRF has multiple branches of residual blocks, each using different kernel sizes to represent various receptive field patterns. After the branches of the residual blocks are processed in parallel, their outputs are integrated via summation. This parallel representation using multiple branches is a crucial factor in high-fidelity speech synthesis with few-channel convolutions. Specifically, HiFi-GAN V2 (a lightweight variant) achieves high speech quality with 128 channels in the first block, which is significantly smaller than that of other models with a similar upsampling scheme (e.g., 512 in MelGAN [20]).

As an example, we present the detailed MRF network architecture in Figure 2(a). In this network, the kernel sizes were set to 3, 7, and 11 in the left, center, and right branches, respectively. The dilation rates were set to 1, 3, and 5 in the first layer of the first, second, and third blocks, respectively. The number of parameters[2] is calculated as follows: $C_{\text{in}} \times C_{\text{out}} \times k \times D = C \times C \times (3+7+11) \times 6 = 126C^2$, where C_{in}, C_{out}, k, and D denote the number of input channels, number of output channels, kernel size, and number of layers, respectively. Specifically, C was set to 128 in HiFi-GAN V2.

2.2. Proposal: Multi-input single shared residual block

MRF enlarges the description capacity by increasing the number of branches of the residual blocks. However, this increases the model size. Therefore, as an alternative, we developed MISR, which is lighter than MRF. MISR expands the description capacity by increasing the input variation using lightweight convolutions with a kernel size of 1. Alternatively, MISR reduces the variation in residual blocks from multiple to single and uses shared-weight residual blocks among the branches. To compensate for this simplification, the outputs of the residual blocks are integrated by applying lightweight convolutions with a kernel size of 1 as opposed to summation. We discuss the effects of this replacement in Section 3.2. Because the model sizes of the input and output convolutions are significantly smaller than those of the residual blocks, MISR reduces the model size compared with that of MRF.

As an example, we provide the detailed network architecture of MISR in Figure 2(b). To ensure that the maximum size

[2]For simplicity, we ignore the parameters for the bias term.

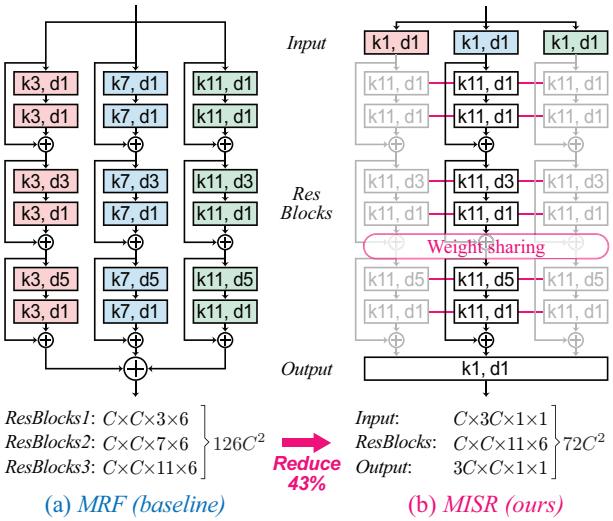

Figure 2: *Network architectures of MRF and MISR. For simplicity, we only present convolution layers with "kx, dy," where x and y indicate the kernel size and dilation rate, respectively.*

of the receptive field is the same as that of MRF, MISR uses convolutions with a kernel size of 11, which are the same as those in the right branch of MRF. The number of parameters[2] is calculated as $C \times 3C \times 1 \times 1 + C \times C \times 11 \times 6 + 3C \times C \times 1 \times 1 = 72C^2$, where the first, second, and third terms represent the numbers of parameters in the input convolutions, residual blocks, and output convolutions, respectively. As discussed in Section 2.1, the number of MRF parameters is $126C^2$; therefore, MISR can reduce the model size by 43%.

2.3. Implementation technique for accelerating MISR

Naive implementation (Algorithm 1). When implementing MISR naively, its processing speed is slower than that of MRF because input and output convolutions are additionally used, whereas the number of branches of the residual blocks applied in the process remains the same (e.g., three in Figure 2). We provide the pseudocode for the naive implementation of MISR in Algorithm 1. Here, the shape of the tensor (e.g., (N, C, L)) is presented following the triangular mark ($\triangleright$), where N, C, L, and b indicate the batch size, number of channels, sequence length, and number of branches, respectively. As described in line 4, the residual blocks are applied b times using the for loop in the naive implementation. This increased computation is a defect in the naive implementation of MISR.

Fast implementation (Algorithm 2). In MISR, the type of residual blocks is unified into a single type. In this case, the processing speed can be improved by applying a typical fast batchwise operation after arranging each branch in a batch dimension. The pseudocode for the fast implementation of MISR is provided in Algorithm 2. We perform tensor reshaping before applying the residual blocks and move the element regarding the branches (b) from the channel (second) dimension to the batch (first) dimension (line 2). By conducting this process, we can simultaneously apply residual blocks to all branches using a typical fast batchwise operation (line 3).

The difference in inference speed between the naive and fast implementations of MISR is listed in Table 1. Herein, we report the results of incorporating MISR into HiFi-GAN V2 [22] and iSTFTNet V2-`C8C8I` [26]. The results indicate

Algorithm 1 Naive implementation of MISR

Input: f_{in} ▷ (N, C, L)
Output: f_{out} ▷ (N, C, L)
$f_1 \leftarrow \text{Conv}_{\text{in}}(f_{\text{in}})$ ▷ (N, bC, L) // Channel expansion
$[f_2^1, \ldots, f_2^b] \leftarrow \text{Split}(f_1)$ ▷ (N, C, L) // Channel split
for $i = 1$ to b **do**
  $f_3^i \leftarrow \text{ResBlocks}(f_2^i)$ ▷ (N, C, L) // Used b times
end for
$f_4 \leftarrow \text{Concat}([f_3^1, \ldots, f_3^b])$ ▷ (N, bC, L) // Channel concatenation
$f_{\text{out}} \leftarrow \text{Conv}_{\text{out}}(f_4)$ ▷ (N, C, L) // Channel reduction

Algorithm 2 Fast implementation of MISR

Input: f_{in} ▷ (N, C, L)
Output: f_{out} ▷ (N, C, L)
$f_1 \leftarrow \text{Conv}_{\text{in}}(f_{\text{in}})$ ▷ (N, bC, L) // Channel expansion
$f_2 \leftarrow \text{Reshape}(f_1)$ ▷ (Nb, C, L) // Channel to batch
$f_3 \leftarrow \text{ResBlocks}(f_2)$ ▷ (Nb, C, L) // Used once
$f_4 \leftarrow \text{Reshape}(f_3)$ ▷ (N, bC, L) // Batch to channel
$f_{\text{out}} \leftarrow \text{Conv}_{\text{out}}(f_4)$ ▷ (N, C, L) // Channel reduction

that the fast implementation technique is valuable for accelerating MISR. Based on these results, we report the results with a fast implementation in subsequent experiments. The experimental setup is described in detail in the following section.

3. Experiments

3.1. Experiment setup

Dataset. To investigate the effectiveness of *MISRNets*, we conducted experiments using the LJSpeech dataset [32], which includes 13,100 audio clips (24 h) of an English female speaker, and 12,600, 250, and 250 audio clips 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 from the audio clips with an FFT size of 1024, hop length of 256, and window length of 1024.

Network architectures. We applied our ideas to two lightweight neural vocoders: HiFi-GAN V2 [22] and iSTFTNet V2-`C8C8I` [26]. iSTFTNet V2-`C8C8I` is a faster and more lightweight variant of HiFi-GAN V2, in which the two output-side residual blocks of the HiFi-GAN V2 generator are replaced with the inverse short-time Fourier transform (iSTFT). We implemented *MISRNets* by replacing the MRF (Figure 2(a)) in HiFi-GAN V2 and iSTFTNet V2-`C8C8I` with MISR (Figure 2(b)). For simplicity, we hereafter denote HiFi-GAN V2 and iSTFTNet V2-`C8C8I` as *HiFi* and *iSTFT*, and their *MISRNet* variants as *HiFi-MISR* and *iSTFT-MISR*, respectively. All models were implemented based on open-source code[3] for easy comparison with the various synthesis speeches provided.

Training settings. We trained the models using the HiFi-GAN configurations provided in the open-source code,[3] where the hyperparameters were tuned for stable training across various datasets. Specifically, we trained the models for 2.5M iterations using the Adam optimizer [33] with a batch size of 16, 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 combined least-squares GAN [34], mel-spectrogram [22], and feature-matching [35, 20] losses.

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

Table 1: *Comparison of inference speed between naive and fast implementations of MISR when incorporated into HiFi-GAN V2 and iSTFTNet V2-*`C8C8I`*. We report the relative speed compared to the real-time audio playing speed when a 1 s audio is processed. The larger the value, the higher the speed.*

Model	Speed↑ (GPU)	Speed↑ (CPU)
HiFi-GAN w/ naive MISR	×116.06	×22.15
HiFi-GAN w/ fast MISR	×**220.63**	×**24.87**
iSTFTNet w/ naive MISR	×190.64	×37.56
iSTFTNet w/ fast MISR	×**358.28**	×**44.09**

Evaluation metrics. We conducted a mean opinion score (*MOS*) test to evaluate perceptual quality. We randomly selected 20 utterances from the evaluation set and used the mel spectrograms extracted from them as vocoder input. The test was conducted online, and 14 listeners participated. The listeners were asked to judge speech quality using five options: 1 = bad, 2 = poor, 3 = fair, 4 = good, and 5 = excellent. Audio samples are available from the link presented on the first page.[1] As an objective metric for perceptual quality, we used the conditional Fréchet wav2vec distance (*cFW2VD*) [26], which calculates the distance between real and generative distributions in a wav2vec 2.0 [36] feature space conditioned on text. This metric is conceptually similar to Fréchet inception distance (FID) [37] and Fréchet DeepSpeech distance (FDSD) [38], which assess the perceptual quality of images and speech, respectively. We used cFW2VD because it is highly correlated with MOS (Spearman's rank correlation of -0.93) [26]. The *smaller* the value, the *better* was the perceptual quality. For the inference speed, we measured the *relative speed* compared to the real-time audio playing speed when a 1 s audio was processed. The speed on the GPU was computed on a single NVIDIA P100 GPU, and that on the CPU was measured on a MacBook Pro with a 2.7 GHz Intel Core i7. The *larger* the value, the *higher* was the speed. For model size, we report the *number of parameters*. The *smaller* the value, the *more lightweight* was the model.

3.2. Results

We examined the proposed model from four perspectives.

(1) Comparison between MRF and MISR. First, we examined the effect of replacing MRF with MISR. Table 2 summarizes the results. We found that *HiFi-MISR*/*iSTFT-MISR* was comparable to the original *HiFi*/*iSTFT* in terms of speech quality (MOS and cFW2VD), whereas the former significantly reduced the model size compared with the latter. For inference speed, *HiFi-MISR*/*iSTFT-MISR* was comparable to *HiFi*/*iSTFT* on the CPU, whereas the former outperformed the latter on the GPU. The GPU inference speed of *HiFi-MISR*/*iSTFT-MISR* with a naive implementation (Table 1) was comparable to that of *HiFi*/*iSTFT* (Table 2). These results indicate that GPU inference speed was improved by the fast implementation of MISR (Algorithm 2).

(2) Importance of multi-input. In a study of MRF [22], it was shown that MRF outperformed a single-receptive field model. To confirm the validity of this result for MISR, we investigated the difference in performance between MISR and a single-input single residual block (*SISR*), where the three branches in MISR (Figure 2(b)) are reduced to a single branch. The results are listed in Table 3. We found that *HiFi-SISR*/*iSTFT-SISR* can improve the inference speed by reducing the number of branches; however, the speech quality deteriorated in terms of MOS and

Table 2: *Comparison of MOS with 95% confidence intervals, cFW2VD, inference speed, and number of parameters between HiFiGAN/iSTFTNet with MRF and that with MISR. The underlined models are MISRNets (lightweight models).*

Model	MOS↑	cFW2VD↓	Speed↑ (GPU)	Speed↑ (CPU)	# Param↓ (M)
Ground truth	4.76 $\pm$0.06	–	–	–	–
HiFi	4.09 $\pm$0.10	0.046	×125.27	×26.31	0.93
HiFi-MISR	4.07 $\pm$0.10	0.047	×220.63	×24.87	0.63
iSTFT	4.17 $\pm$0.09	0.042	×192.56	×45.92	0.89
iSTFT-MISR	4.16 $\pm$0.09	0.046	×358.28	×44.09	0.61

cFW2VD. The results indicate that MISR is more appropriate than SISR for obtaining a lightweight model while retaining the speech quality.

Table 3: *Comparison of MOS with 95% confidence intervals, cFW2VD, inference speed, and number of parameters between multi-input (MISR) and single-input (SISR) models.*

Model	MOS↑	cFW2VD↓	Speed↑ (GPU)	Speed↑ (CPU)	# Param↓ (M)
HiFi-MISR	4.07 $\pm$0.10	0.047	×220.63	×24.87	0.63
HiFi-SISR	3.91 $\pm$0.11	0.069	×262.19	×59.05	0.61
iSTFT-MISR	4.16 $\pm$0.09	0.046	×358.28	×44.09	0.61
iSTFT-SISR	3.73 $\pm$0.11	0.071	×368.91	×90.22	0.59

(3) Importance of output convolutions. As discussed in Section 2.2, MISR integrates the outputs of the residual blocks using convolutions instead of summation to compensate for the simplification of the variation in the residual blocks. To confirm the importance of this replacement, we examined the difference in performance between MISR and MISR without output convolutions (summation was alternatively used). We denote this model as *MISR*†. Table 4 summarizes the results. We found that the improvement in inference speed and model size by replacing *MISR* with *MISR*† was subtle because the output convolutions with a kernel size of 1 are lightweight and fast. For speech quality (MOS and cFW2VD), *HiFi-MISR*†/*iSTFT-MISR*† performed more poorly than *HiFi-MISR*/*iSTFT-MISR*. These results verify the importance of using output convolutions in MISR.

Table 4: *Comparison of MOS with 95% confidence intervals, cFW2VD, inference speed, and number of parameters between MISR and MISR without output convolutions (MISR*†*).*

Model	MOS↑	cFW2VD↓	Speed↑ (GPU)	Speed↑ (CPU)	# Param↓ (M)
HiFi-MISR	4.07 $\pm$0.10	0.047	×220.63	×24.87	0.63
HiFi-MISR†	3.97 $\pm$0.10	0.064	×233.10	×25.08	0.62
iSTFT-MISR	4.16 $\pm$0.09	0.046	×358.28	×44.09	0.61
iSTFT-MISR†	4.07 $\pm$0.10	0.062	×371.07	×44.73	0.60

(4) Comparison with other lightweight modules. We must determine whether MISR is reasonable for reducing model size. Thus, we investigated the difference in performance between MISR and depthwise separable convolution (*DSC*) [28, 29, 30, 31], which is a commonly used module for model size reduction. For a fair comparison, we adjusted the parameters of DSC (more concretely, the depth multiplier) such that its model size was comparable to that of MISR. Table 5 presents the results. We found that *HiFi-DSC*/*iSTFT-DSC* can reduce the model size, similar to *HiFi-MISR*/*iSTFT-MISR*, but cannot improve the GPU inference speed compared with *HiFi*/*iSTFT* (Table 2), in contrast to *HiFi-MISR*/*iSTFT-MISR*. Note that we implemented the models in PyTorch [39], and there is scope for improvement by applying further optimization. Regarding speech quality (MOS and cFW2VD), *HiFi-DSC*/*iSTFT-DSC* performed more poorly than *HiFi-MISR*/*iSTFT-MISR*. Therefore, MISR is more appropriate for reducing the model size while retaining speech quality. Notably, MISR and DSC are compatible and can be used together for further model size reduction, which will be the subject of future research.

Table 5: *Comparison of MOS with 95% confidence intervals, cFW2VD, inference speed, and number of parameters between lightweight modules (i.e., MISR and DSC).*

Model	MOS↑	cFW2VD↓	Speed↑ (GPU)	Speed↑ (CPU)	# Param↓ (M)
HiFi-MISR	4.07 $\pm$0.10	0.047	×220.63	×24.87	0.63
HiFi-DSC	3.52 $\pm$0.12	0.061	×95.29	×24.58	0.59
iSTFT-MISR	4.16 $\pm$0.09	0.046	×358.28	×44.09	0.61
iSTFT-DSC	3.86 $\pm$0.11	0.061	×179.54	×45.54	0.56

3.3. Application to multi-speaker and Japanese datasets

To examine the generality of *MISRNet*, we examined its performance when applied to multi-speaker and Japanese datasets. As a multi-speaker dataset, we used the VCTK dataset [40], with 44,200 audio clips (44 h) of 109 English speakers. For the Japanese dataset, we used the JSUT dataset [41], which includes 7,696 audio clips (10 h) from a Japanese female speaker. We implemented the models based on open-source code.[3] We provide cFW2VD in Table 6. We found that *HiFi-MISR/iSTFT-MISR* was comparable to *HiFi/iSTFT* in terms of cFW2VD on these datasets. The results do not contradict those in Section 3.2. We have provided audio samples on the website presented on the first page.[1]

Table 6: *Comparison of cFW2VD between HiFiGAN/iSTFTNet with MRF and that with MISR on the VCTK and JSUT datasets.*

Model	cFW2VD↓ on VCTK	cFW2VD↓ on JSUT
HiFi	0.065	0.065
HiFi-MISR	0.070	0.065
iSTFT	0.067	0.070
iSTFT-MISR	0.070	0.069

4. Conclusion

We developed *MISRNet* to make two lightweight neural vocoders (i.e., HiFi-GAN and iSTFTNet) further lightweight while retaining speech quality. Furthermore, we introduced a fast implementation of MISR to accelerate inference speed. The experimental results demonstrated that the presented techniques are essential and sufficient for achieving these objectives. Although this study focused on HiFi-GAN, MISR is a general concept that can be applied to various models beyond HiFi-GAN; this remains the subject of future research.

5. Acknowledgements

This work was partially supported by JST CREST Grant Number JPMJCR19A3, Japan.

6. References

[1] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerrv-Ryan, R. A. Saurous, Y. Agiomyrgiannakis, and Y. Wu, "Natural TTS synthesis by conditioning WaveNet on mel spectrogram predictions," in *Proc. ICASSP*, 2018, pp. 4779–4783.

[2] W. Ping, K. Peng, A. Gibiansky, S. O. Arik, A. Kannan, S. Narang, J. Raiman, and J. Miller, "Deep Voice 3: Scaling text-to-speech with convolutional sequence learning," in *Proc. ICLR*, 2018.

[3] N. Li, S. Liu, Y. Liu, S. Zhao, and M. Liu, "Neural speech synthesis with transformer network," in *Proc. AAAI*, 2019, pp. 6706–6713.

[4] Y. Ren, Y. Ruan, X. Tan, T. Qin, S. Zhao, Z. Zhao, and T.-Y. Liu, "FastSpeech: Fast, robust and controllable text to speech," in *Proc. NeurIPS*, 2019, pp. 3171–3180.

[5] K. Qian, Y. Zhang, S. Chang, X. Yang, and M. Hasegawa-Johnson, "Auto-VC: Zero-shot voice style transfer with only autoencoder loss," in *Proc. ICML*, 2019, pp. 5210–5219.

[6] T. Kaneko, H. Kameoka, K. Tanaka, and N. Hojo, "CycleGAN-VC3: Examining and improving CycleGAN-VCs for mel-spectrogram conversion," in *Proc. Interspeech*, 2020, pp. 2017–2021.

[7] ——, "MaskCycleGAN-VC: Learning non-parallel voice conversion with filling in frames," in *Proc. ICASSP*, 2021, pp. 5919–5923.

[8] H. Kameoka, K. Tanaka, and T. Kaneko, "FastS2S-VC: Streaming non-autoregressive sequence-to-sequence voice conversion," *arXiv preprint arXiv:2104.06900*, 2021.

[9] 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 preprint arXiv:1609.03499*, 2016.

[10] N. Kalchbrenner, E. Elsen, K. Simonyan, S. Noury, N. Casagrande, E. Lockhart, F. Stimberg, A. van den Oord, S. Dieleman, and K. Kavukcuoglu, "Efficient neural audio synthesis," in *Proc. ICML*, 2018, pp. 2410–2419.

[11] A. van den Oord, Y. Li, I. Babuschkin, K. Simonyan, O. Vinyals, K. Kavukcuoglu, G. van den Driessche, E. Lockhart, L. Cobo, F. Stimberg, N. Casagrande, D. Grewe, S. Noury, S. Dieleman, E. Elsen, N. Kalchbrenner, H. Zen, A. Graves, H. King, T. Walters, D. Belov, and D. Hassabis, "Parallel WaveNet: Fast high-fidelity speech synthesis," in *Proc. ICML*, 2018, pp. 3918–3926.

[12] W. Ping, K. Peng, and J. Chen, "ClariNet: Parallel wave generation in end-to-end text-to-speech," in *Proc. ICLR*, 2019.

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

[14] D. P. Kingma and P. Dhariwal, "Glow: Generative flow with invertible 1×1 convolutions," in *Proc. NeurIPS*, 2018, pp. 10 236–10 245.

[15] N. Chen, Y. Zhang, H. Zen, R. J. Weiss, M. Norouzi, and W. Chan, "WaveGrad: Estimating gradients for waveform generation," in *Proc. ICLR*, 2020.

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

[17] Y. Song and S. Ermon, "Generative modeling by estimating gradients of the data distribution," in *Proc. NeurIPS*, 2019, pp. 11 918–11 930.

[18] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *Proc. NeurIPS*, 2020, pp. 6840–6851.

[19] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. NIPS*, 2014, pp. 2672–2680.

[20] K. Kumar, R. Kumar, T. de Boissiere, L. Gestin, W. Z. Teoh, J. Sotelo, A. de Brébisson, Y. Bengio, and A. Courville, "MelGAN: Generative adversarial networks for conditional waveform synthesis," in *Proc. NeurIPS*, 2019, pp. 14 910–14 921.

[21] 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.

[22] J. Kong, J. Kim, and J. Bae, "HiFi-GAN: Generative adversarial networks for efficient and high fidelity speech synthesis," in *Proc. NeurIPS*, 2020, pp. 17 022–17 033.

[23] J. Yang, J. Lee, Y. Kim, H. Cho, and I. Kim, "VocGAN: A high-fidelity real-time vocoder with a hierarchically-nested adversarial network," in *Proc. Interspeech*, 2020, pp. 200–204.

[24] 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*, 2021, pp. 492–498.

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

[26] 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*, 2022, pp. 6207–6211.

[27] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. CVPR*, 2016, pp. 770–778.

[28] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proc. CVPR*, 2017, pp. 1251–1258.

[29] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[30] B. Zhai, T. Gao, F. Xue, D. Rothchild, B. Wu, J. E. Gonzalez, and K. Keutzer, "SqueezeWave: Extremely lightweight vocoders for on-device speech synthesis," *arXiv preprint arXiv:2001.05685*, 2020.

[31] B. Wu, Q. He, P. Zhang, T. Koehler, K. Keutzer, and P. Vajda, "FBWave: Efficient and scalable neural vocoders for streaming text-to-speech on the edge," *arXiv preprint arXiv:2011.12985*, 2020.

[32] K. Ito and L. Johnson, "The LJ speech dataset," https://keithito.com/LJ-Speech-Dataset/, 2017.

[33] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. ICLR*, 2015.

[34] X. Mao, Q. Li, H. Xie, R. Y. Lau, Z. Wang, and S. P. Smolley, "Least squares generative adversarial networks," in *Proc. ICCV*, 2017, pp. 2794–2802.

[35] A. B. L. Larsen, S. K. Sønderby, H. Larochelle, and O. Winther, "Autoencoding beyond pixels using a learned similarity metric," in *Proc. ICML*, 2016, pp. 1558–1566.

[36] A. Baevski, H. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," in *Proc. NeurIPS*, 2020, pp. 12 449–12 460.

[37] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "GANs trained by a two time-scale update rule converge to a local Nash equilibrium," in *Proc. NIPS*, 2017, pp. 6629–6640.

[38] M. Bińkowski, J. Donahue, S. Dieleman, A. Clark, E. Elsen, N. Casagrande, L. C. Cobo, and K. Simonyan, "High fidelity speech synthesis with adversarial networks," in *Proc. ICLR*, 2020.

[39] 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*, 2019, pp. 8026–8037.

[40] V. Christophe, Y. Junichi, and M. Kirsten, "CSTR VCTK corpus: English multi-speaker corpus for CSTR voice cloning toolkit," *The Centre for Speech Technology Research*, 2016.

[41] R. Sonobe, S. Takamichi, and H. Saruwatari, "JSUT corpus: free large-scale japanese speech corpus for end-to-end speech synthesis," *arXiv preprint arXiv:1711.00354*, 2017.