INTERSPEECH 2023
20-24 August 2023, Dublin, Ireland

iSTFTNet2: Faster and More Lightweight iSTFT-Based Neural Vocoder Using 1D-2D CNN

Takuhiro Kaneko, Hirokazu Kameoka, Kou Tanaka, Shogo Seki

NTT Communication Science Laboratories, NTT Corporation, Japan

takuhiro.kaneko@ntt.com

Abstract

The inverse short-time Fourier transform network (iSTFTNet) has garnered attention owing to its fast, lightweight, and high-fidelity speech synthesis. It obtains these characteristics using a fast and lightweight 1D CNN as the backbone and replacing some neural processes with iSTFT. Owing to the difficulty of a 1D CNN to model high-dimensional spectrograms, the frequency dimension is reduced via temporal upsampling. However, this strategy compromises the potential to enhance the speed. Therefore, we propose *iSTFTNet2*, an improved variant of iSTFTNet with a 1D-2D CNN that employs 1D and 2D CNNs to model temporal and spectrogram structures, respectively. We designed a 2D CNN that performs frequency upsampling after conversion in a few-frequency space. This design facilitates the modeling of high-dimensional spectrograms without compromising the speed. The results demonstrated that iSTFTNet2 made iSTFTNet faster and more lightweight with comparable speech quality.[1]

Index Terms: speech synthesis, neural vocoder, inverse short-time Fourier transform, convolutional neural network, generative adversarial networks

1. Introduction

Text-to-speech (TTS) synthesis and voice conversion (VC) have been extensively studied to obtain the desired speech. The two-stage approach widely used in TTS and VC is as follows. The first model predicts the intermediate representation (e.g., mel-spectrogram) from the input data (e.g., text or speech), whereas the second model synthesizes speech from the predicted intermediate representation. This study focuses on the second model, the neural vocoder, and attempts to make it faster and more lightweight to broaden its applicability.

Various neural vocoders have been developed with advances in deep generative models. The pioneer is an autoregressive model (e.g., WaveNet [1] and WaveRNN [2]) that achieves high-fidelity speech synthesis but suffers from slow inference owing to sample-by-sample processing. Various parallelizable non-autoregressive models have been developed to boost the inference speed. For example, successful models include a distillation-based (e.g., Parallel WaveNet [3] and ClariNet [4]), flow (e.g., Glow [5])-based (e.g., WaveGlow [6]), diffusion probabilistic model [7, 8]-based (e.g., WaveGrad [9] and DiffWave [10]), and generative adversarial network (GAN) [11]-based (e.g., [12–24]) models. Among them, this study focuses on a GAN-based model while prioritizing the flexibility of the architectural design and the ability of fast inference.

[1] Audio samples are available at https://www.kecl.ntt.co.jp/people/kaneko.takuhiro/projects/istftnet2/.

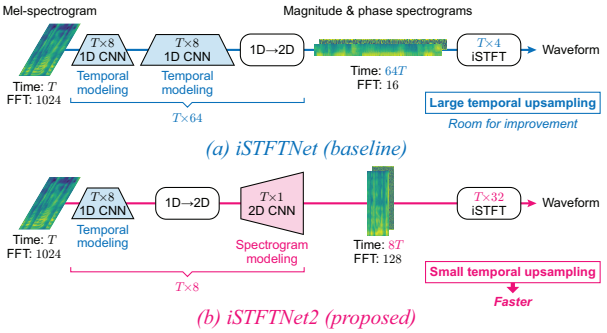


Figure 1: *Comparison of iSTFTNet [20] and iSTFTNet2 (proposed). (a) Owing to the difficulty of a 1D CNN to model high-dimensional spectrograms, it is necessary to reduce the frequency dimension sufficiently in iSTFTNet using large temporal upsampling. (b) In contrast, iSTFTNet2 mitigates this difficulty by conducting 1D-to-2D conversion in an earlier stage and applying a 2D CNN that can capture local structures in spectrograms. This modification facilitates the reduction of the neural temporal upsampling by eight times (i.e., from* $\times 64$ *to* $\times 8$*) and enhances the inference speed.*

Among the GAN-based neural vocoders, one of the fastest and most lightweight models is the inverse short-time Fourier transform network (iSTFTNet) [20], which achieves fast, lightweight, and high-fidelity speech synthesis using a fast and lightweight 1D CNN (e.g., HiFi-GAN [14]) as the backbone and replacing some output-side neural processes with fast and lightweight inverse short-time Fourier transform (iSTFT). Particularly, iSTFTNet applies iSTFT after sufficiently reducing the frequency dimension using large temporal upsampling (Figure 1(a)) to avoid modeling high-dimensional spectrograms, which are difficult for a 1D CNN to represent. This technique is essential for making the model faster and more lightweight while maintaining speech quality; however, it compromises the potential to improve the inference speed by applying iSTFT with fewer temporal upsampling.

One possible solution is to conduct spectrogram conversion using a fully 2D CNN (e.g., [25–27]). However, the direct application of a 2D CNN requires a significant increase in the calculation cost because it increases linearly according to the frequency dimension (e.g., 80 in a mel-spectrogram). Alternatively, inspired by the success of combining 1D and 2D CNNs [28], we propose *iSTFTNet2*, an improved variant of iSTFTNet with a 1D-2D CNN, in which 1D and 2D CNNs are used to model the global temporal and local spectrogram structures, respectively (Figure 1(b)). Particularly, we designed a 2D CNN that conducts frequency upsampling after performing

10.21437/Interspeech.2023-1726

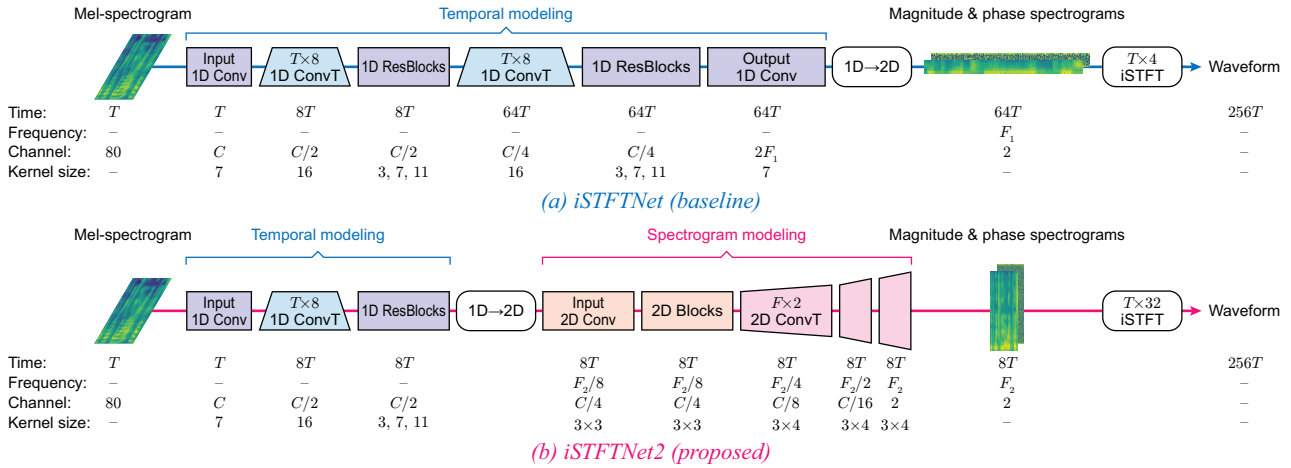


Figure 2: *Overall architectures of iSTFTNet [20] and iSTFTNet2 (proposed) when incorporated into HiFi-GAN [14]. T and C denote the temporal length and number of output channels, respectively. F_1 indicates the frequency dimension of the spectrogram in iSTFTNet, whereas F_2 indicates that in iSTFTNet2. C is set to 128 when HiFi-GAN V2 (lightweight variant) is used as the backbone. F_1 and F_2 can be calculated by $\frac{f_s}{2}+1$ (see Equation 1 for the definition of f_s). When the FFT size is 1024, they are calculated as $F_1 = \frac{1024/64}{2}+1 = 9$ and $F_1 = \frac{1024/8}{2}+1 = 65$.*

sufficient conversion in a few-frequency space using 1D and few-frequency 2D CNNs. This design facilitates the modeling of high-dimensional spectrograms, which are difficult for a conventional 1D CNN-based iSTFTNet to model, without compromising the speed. Furthermore, we propose an efficient module inspired by ShuffleNets [29, 30] to improve the speed and weight of the model further.

In the experiments, we examined the effectiveness of iSTFTNet2 on two representative datasets: LJSpeech [31] (single English speaker) and VCTK [32] (multiple English speakers). The experimental results demonstrated that iSTFTNet2 made iSTFTNet faster and more lightweight with comparable speech quality. Furthermore, we demonstrated the versatility of our ideas by applying iSTFTNet2 to multi-band modeling [16, 33], another technique for improving the speed. The results showed that this variant could further improve the speed with comparable speech quality.

The remainder of this paper is organized as follows. Section 2 briefly reviews the conventional iSTFTNet. Section 3 presents details of the proposed iSTFTNet2. Section 4 presents our experimental results. Finally, Section 5 concludes the study and discusses future research.

2. Preliminary: Conventional iSTFTNet

iSTFTNet [20] is one of the fastest and most lightweight available neural vocoders. These characteristics were obtained by replacing some of the output-side layers of a fully neural vocoder with fast and lightweight iSTFT. Particularly, it uses a fast and lightweight 1D CNN (e.g., HiFi-GAN [14]) as the backbone with a high processing speed. However, it is challenging for a 1D CNN to model high-dimensional spectrograms because of the difficulty in capturing local structures in the frequency direction. Hence, iSTFTNet reduces the frequency dimension using temporal upsampling as follows:

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

where f_s, h_s, and w_s indicate the FFT size, hop length, and window length, respectively, required for the iSTFT after $\times s$ temporal upsampling. This equation is based on the time and frequency tradeoff, that is, $f_1 \cdot 1 = f_s \cdot s =$ constant and indicates that the frequency dimension can be reduced s times by conducting $\times s$ temporal upsampling.

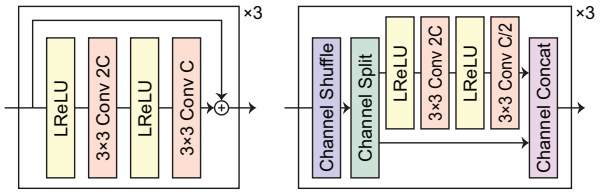
Figure 2(a) shows the overall architecture of iSTFTNet. We present the architecture of iSTFTNet-`C8C8I4`,[2] which is the best balanced model that improves the speed and model size while maintaining speech quality, where `C`x indicates the use of 1D CNN blocks with $\times x$ temporal upsampling, and `I`y indicates the use of iSTFT with $\times y$ temporal upsampling. When prioritizing the speed, a model that performs temporal upsampling fewer times, for example, iSTFTNet-`C8C1I32`, which conducts temporal upsampling once, is better. However, it is shown that such a model deteriorates speech quality because of the difficulty of a 1D CNN in modeling high-dimensional spectrograms [20].

3. Proposal: iSTFTNet2

Considering the abovementioned facts, we attempted to construct an improved variant of iSTFTNet that can maintain speech quality even with fewer temporal upsampling. A possible solution is to convert a spectrogram using a fully 2D CNN that can capture the local structures in spectrograms (e.g., [25–27]). However, this replacement requires a significant increase in the calculation cost because it increases linearly in proportion to the frequency dimension (e.g., 80 in a mel-spectrogram).

Alternatively, we developed *iSTFTNet2*, which constitutes a 1D-2D CNN. Figure 2(b) presents the overall architecture of iSTFTNet2. As shown in this figure, to model temporal structures efficiently, iSTFTNet2 uses the same 1D CNN for the first three modules as that used in iSTFTNet, except that channel concatenation is used instead of addition when integrating the outputs of the multi-receptive fusion [14] in the 1D ResBlock to propagate more information to the subsequent 2D CNN. Unlike iSTFTNet, iSTFTNet2 conducts 1D-to-2D conversion in an earlier step and applies a 2D CNN to effectively capture the lo-

[2]This is the same as iSTFTNet-`C8C8` described in [20]. Here, we added `I4` to specify the temporal upsampling scale in the iSTFT.



(a) 2D ResBlocks *(b) 2D ShuffleBlocks*

Figure 3: *Architectures of 2D blocks used in iSTFTNet2. "LReLU" denotes a leaky rectified linear unit [34] with a negative slope of* 0.1. *"$x \times y$ Conv z" indicates a 2D convolution with a kernel size of $x \times y$ and the number of output channels of z, and C in this module denotes the number of input channels of the block. In (a), residual connection (which is denoted by "+") [35] is used. In (b), the weight-free operations used in ShuffleNets [29, 30], i.e., "Channel Shuffle," "Channel Split," and "Channel Concat," are used. In both blocks, the presented block is stacked three times.*

cal structures in the spectrograms, which are difficult for a 1D CNN to model.

When considering the detailed configuration of a 2D CNN, it is important to prevent an increase in the calculation cost driven by the introduction of a 2D CNN because its calculation cost increases linearly in proportion to the time and frequency dimensions. To address this problem, iSTFTNet2 performs the main conversion in a few-frequency space (specifically, 2D blocks are applied in a space in which the frequency dimension is downsampled eight times, as shown in Figure 2(b)), and then conducts frequency upsampling in the last phase using transposed convolutions.

The design of the 2D blocks is a vital aspect to consider. As shown in Figure 3, we developed two architectural designs. The first is a 2D ResBlock (Figure 3(a)) that uses a residual connection [35] to propagate information efficiently. We adjusted the model parameters (i.e., number of channels and kernel size) such that the model became faster and more lightweight than iSTFTNet-`C8C8I4` (the best-balanced model). To further make the model faster and more lightweight, we introduced a second model, that is, a 2D ShuffleBlock (Figure 3(b)), which is inspired by efficient neural networks called ShuffleNets [29, 30]. In this block, the number of weight parameters used in the 2D convolutional layers was adjusted such that it was half of that of the 2D ResBlock (Figure 3(a)). Alternatively, in contrast to the residual connection, the half channels are propagated directly without any addition to preserve the model capacity. A channel shuffle [29, 30] is conducted to provide an interaction between the skip and non-skip branches. Because the channel shuffle, channel split, and channel concat are weight-free operations, this block is faster and more lightweight than the 2D ResBlock. We demonstrated the empirical performance difference between the two 2D blocks in the experiments presented in the next section.

4. Experiments

4.1. Experimental setup

Dataset. We examined the effectiveness of iSTFTNet2 using two representative datasets. *LJSpeech* [31], which includes 13,100 audio clips of a single English speaker, and 12,500, 100, and 500 audio clips were used for training, validation, and evaluation, respectively. *VCTK* [32], which comprises 44,081 audio clips of 108 different English speakers, and 41,921, 1,080, and 1,080 audio clips were used for training, validation, and evaluation, respectively. Following a study on HiFi-GAN [14], 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, a hop length of 256, and a window length of 1024.

Implementation. We used *iSTFTNet-`C8C8I4`* [20] as a baseline because it is the best-balanced model and its speech quality has been demonstrated [20] to be comparable to that of HiFi-GAN [14], which is a widely used baseline model. Specifically, we implemented iSTFTNet-`C8C8I4` based on *HiFi-GAN V2* (a lightweight variant) because we were interested in the performance of the lightweight model.[3] We implemented iSTFTNet2 by replacing the modules of iSTFTNet-`C8C8I4`, as shown in Figure 2. Particularly, we implemented the two variants using different 2D blocks, as shown in Figure 3. For clarity, we denote iSTFTNet2 with 2D ResBlocks (Figure 3(a)) and that with 2D ShuffleBlocks (Figure 3(b)) as *iSTFTNet2-Base* and *iSTFTNet2-Small*, respectively. As another comparison model, we examined *iSTFTNet-`C8C1I32`*, which conducts the same temporal upsampling as iSTFTNet2 but uses 1D ResBlocks instead of 2D blocks unlike iSTFTNet2. This model was used to validate the importance of 2D blocks. All models were implemented based on open-source code,[4] and the same training settings were used. Specifically, a combination of least-squares GAN [36], mel-spectrogram [14], and feature matching [12,37] losses was used as the loss function. Each model was trained for 2.5M iterations using the Adam optimizer [38] with a batch size of 16, an initial learning rate of 0.0002, and momentum terms β_1 and β_2 of 0.5 and 0.9, respectively.

Evaluation metrics. We conducted mean opinion score (*MOS*) tests to evaluate perceptual quality. Twenty audio clips were randomly selected from the evaluation set, and log-mel spectrograms extracted from the audio clips were used as vocoder inputs. In addition to the speech synthesized by the abovementioned models, *ground-truth* speech was included as an anchor sample. Ten listeners participated in each online test and were asked to assess speech quality using a five-grade evaluation: 1 = bad, 2 = poor, 3 = fair, 4 = good, and 5 = excellent. As an objective metric, we used the conditional Fréchet wav2vec distance (*cFW2VD*) [20], which measures the distribution distance between the real and synthesized speech in a wav2vec 2.0 [39] feature space conditioned to the text. The smaller the value, the better the speech quality. We evaluated the inference speed using a real-time factor (*RTF*) that was calculated by dividing the inference time by the duration of the synthesized speech (fixed at 1 s in the experiments). The *RTF* was measured using a single thread on an Intel Core i7-12700H CPU. The smaller the value, the higher the speed. We evaluated the model size using the number of parameters (*# Param*). The smaller the value, the more lightweight the model. The audio samples are available from the link on the first page.[1]

4.2. Results on single speaker dataset

Table 1 summarizes the results on LJSpeech. These results are discussed from three perspectives.

Speech quality. For the MOS test, we conducted the Mann–Whitney U test. We found that iSTFTNet-`C8C8I4`,

[3] In additional experiments, we also examined the performance when HiFi-GAN V1 (a high-quality variant) was used as a baseline, and observed a similar tendency in terms of cFW2VD, RTF, and # Param.

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

Table 1: *Comparison of MOS with 95% confidence intervals, cFW2VD, RTF, and # Param on LJSpeech. The numbers in () indicate the rates (%) compared with HiFi-GAN V2.*

Model	MOS↑	cFW2VD↓	RTF↓	# Param↓
Ground truth	4.71 ±0.07	–	–	–
HiFi-GAN V2	4.20 ±0.10	0.046	0.053 (100)	0.93M (100)
iSTFTNet-C8C8I4	4.12 ±0.10	0.042	0.029 (55)	0.89M (96)
iSTFTNet-C8C1I32	3.71 ±0.13	0.071	0.018 (34)	1.30M (140)
iSTFTNet2-Base	4.24 ±0.10	0.036	0.021 (41)	0.85M (91)
iSTFTNet2-Small	4.22 ±0.10	0.040	0.018 (35)	0.79M (85)

Table 2: *Comparison of MOS with 95% confidence intervals, cFW2VD, RTF, and # Param on VCTK. The numbers in () indicate the rates (%) compared with HiFi-GAN V2.*

Model	MOS↑	cFW2VD↓	RTF↓	# Param↓
Ground truth	4.38 ±0.09	–	–	–
HiFi-GAN V2	3.99 ±0.11	0.061	0.053 (100)	0.93M (100)
iSTFTNet-C8C8I4	3.94 ±0.12	0.065	0.029 (55)	0.89M (96)
iSTFTNet-C8C1I32	3.40 ±0.13	0.110	0.018 (34)	1.30M (140)
iSTFTNet2-Base	3.91 ±0.11	0.062	0.021 (41)	0.85M (91)
iSTFTNet2-Small	3.91 ±0.12	0.067	0.018 (35)	0.79M (85)

iSTFTNet2-Base, and iSTFTNet2-Small were *not* significantly different from HiFi-GAN V2 in terms of the p-values > 0.05. In contrast, iSTFTNet-C8C8I32 performed significantly worse than the others. cFW2VD was also the worst in iSTFTNet-C8C8I32 and was comparable in the other cases. These results indicate that iSTFTNet2 can be used as an alternative to iSTFTNet and HiFi-GAN regarding speech quality.

Inference speed. The RTF shows that both iSTFTNet2-Base and iSTFTNet2-Small were faster than HiFi-GAN and iSTFTNet-C8C8I4, achieving comparable speech quality. iSTFTNet2-Small was the fastest among them and was comparable to iSTFTNet-C8C8I32, which sacrifices speech quality.

Model size. We found that iSTFTNet2-Base and iSTFTNet2-Small were lighter than all baselines, and iSTFTNet2-Small was the most lightest in terms of # Param.[5]

4.3. Results on multiple speaker dataset

Table 2 lists the results on VCTK. We observed that the same tendency as that observed on LJSpeech. For the MOS test, iSTFTNet-C8C8I4, iSTFTNet2-Base, and iSTFTNet2-Small were *not* significantly different from HiFi-GAN V2 in terms of p-values > 0.05 in the Mann–Whitney U test, whereas iSTFTNet-C8C8I32 performed significantly worse than the others. The RTF and # Param were the same as those observed on LJSpeech.

4.4. Application to multi-band modeling

As discussed in [20], iSTFT and multi-band modeling [16, 33] (another technique for improving speed) are complementary, and the speed can be further enhanced by combining them using iSTFT $\left(\frac{f_1}{sb}, \frac{h_1}{sb}, \frac{w_1}{sb}\right)$, where b is the number of sub-bands. Motivated by this fact, we evaluated the performance of applying iSTFTNet2 to multi-band modeling. We examined the effectiveness of this variant on LJSpeech [31].

[5]# Param of iSTFTNet-C8C1I32 is larger than that of iSTFTNet-C8C8I4 because the number of channels is halved in the second 1D ResBlocks in iSTFTNet-C8C8I4, whereas it is not conducted in iSTFTNet-C8C1I32 owing to the absence of temporal upsampling. We used this strategy to confirm whether iSTFTNet-C8C1I32 could not obtain comparable speech quality, even with expressive modules.

Table 3: *Comparison of MOS with 95% confidence intervals, cFW2VD, RTF, and # Param when incorporating multi-band modeling on LJSpeech. The numbers in () indicate the rates (%) compared with HiFi-GAN V2.*

Model	MOS↑	cFW2VD↓	RTF↓	# Param↓
Ground truth	4.71 ±0.07	–	–	–
iSTFTNet-MB	4.05 ±0.12	0.061	0.012 (22)	0.82M (88)
iSTFTNet2-MB	4.25 ±0.11	0.040	0.011 (21)	0.83M (89)

Implementation. As a baseline, we used iSTFTNet-C4C4I4B4, where C/I/Bx indicates the use of 1D blocks/iSTFT/multi-band modeling with $\times x$ temporal upsampling. We denote this model as *iSTFTNet-MB*. We modified this model to iSTFTNet2 by replacing the second C4 with 2D ShuffleBlocks (Figure 3(b)) and using I16 instead of I4. The number of final output channels of the 2D CNN (2 in Figure 2) was modified to 8 to produce four sub-bands. To allow for this expansion, we doubled the number of channels in the 2D CNN and alternatively changed the number of output channels in the first convolution layer in a 2D ShuffleBlock (Figure 3(b)) from $2C$ to C to make the model size and inference speed similar to those of iSTFTNet-MB. We denote this model as *iSTFTNet2-MB*.

Results. Table 3 lists the results. The RTF and # Param were almost the same for iSTFTNet-MB and iSTFTNet2-MB because we adjusted the model parameters of iSTFTNet2-MB such that they were almost the same. However, iSTFTNet2-MB significantly outperformed iSTFTNet-MB in terms of MOS (with a p-value < 0.05 in the Mann–Whitney U test) and cFW2VD. This is possibly because it is easier to represent multiple sub-band spectrograms simultaneously in a 2D CNN (in which channels and frequencies are represented in independent dimensions) than in a 1D CNN (in which they are mixed in the same dimension). Furthermore, iSTFTNet2-MB was *not* significantly different from HiFi-GAN V2 (Table 1) in these metrics (for the MOS, the p-value > 0.05 in the Mann–Whitney U test), while reducing the RTF to 21%. These results indicated that iSTFTNet2-MB was the best among the variants of iSTFTNets and iSTFTNet2s when prioritizing speed and speech quality.

5. Conclusions

We proposed *iSTFTNet2*, an improved variant of iSTFTNet that constitutes a 1D-2D CNN, in which 1D and 2D CNNs are used to model temporal and spectrogram structures, respectively. The proposed architecture facilitated the application of iSTFT to higher-dimensional spectrograms without large temporal upsampling, and the experimental results demonstrated that iSTFTNet2 made iSTFTNet faster and more lightweight while maintaining speech quality. Although we focused on a GAN-based neural vocoder, our ideas have high applicability, and applying them to other models, including other neural vocoders (e.g., [6, 9, 10]) and end-to-end text-to-speech synthesis (e.g., [4, 40–44]), remains the subject of future research.

6. Acknowledgements

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

7. References

[1] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "WaveNet: A generative model for raw audio," *arXiv preprint arXiv:1609.03499*, 2016.

[2] 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.

[3] 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.

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

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

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

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

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

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

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

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] J.-H. Kim, S.-H. Lee, J.-H. Lee, and S.-W. Lee, "Fre-GAN: Adversarial frequency-consistent audio synthesis," in *Proc. Interspeech*, 2021, pp. 2197–2201.

[19] T. Okamoto, T. Toda, and H. Kawai, "Multi-stream HiFi-GAN with data-driven waveform decomposition," in *Proc. ASRU*, 2021, pp. 610–617.

[20] 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.

[21] S.-H. Lee, J.-H. Kim, K.-E. Lee, and S.-W. Lee, "Fre-GAN 2: Fast and efficient frequency-consistent audio synthesis," in *Proc. ICASSP*, 2022, pp. 6192–6196.

[22] T. Kaneko, H. Kameoka, K. Tanaka, and S. Seki, "MISRNet: Lightweight neural vocoder using multi-input single shared residual blocks," in *Proc. Interspeech*, 2022, pp. 1631–1635.

[23] 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*, 2022, pp. 884–891.

[24] T. Kaneko, H. Kameoka, K. Tanaka, and S. Seki, "Wave-U-Net Discriminator: Fast and lightweight discriminator for generative adversarial network-based speech synthesis," in *Proc. ICASSP*, 2023.

[25] T. Kaneko, S. Takaki, H. Kameoka, and J. Yamagishi, "Generative adversarial network-based postfilter for STFT spectrograms." in *Proc. Interspeech*, 2017, pp. 3389–3393.

[26] K. Oyamada, H. Kameoka, T. Kaneko, K. Tanaka, N. Hojo, and H. Ando, "Generative adversarial network-based approach to signal reconstruction from magnitude spectrogram," in *Proc. EUSIPCO*, 2018, pp. 2514–2518.

[27] P. Neekhara, C. Donahue, M. Puckette, S. Dubnov, and J. McAuley, "Expediting TTS synthesis with adversarial vocoding," in *Proc. Interspeech*, 2019, pp. 186–190.

[28] T. Kaneko, H. Kameoka, K. Tanaka, and N. Hojo, "CycleGAN-VC2: Improved CycleGAN-based non-parallel voice conversion," in *Proc. ICASSP*, 2019, pp. 6820–6824.

[29] X. Zhang, X. Zhou, M. Lin, and J. Sun, "ShuffleNet: An extremely efficient convolutional neural network for mobile devices," in *Proc. CVPR*, 2018, pp. 6848–6856.

[30] N. Ma, X. Zhang, H.-T. Zheng, and J. Sun, "ShuffleNet V2: Practical guidelines for efficient CNN architecture design," in *Proc. ECCV*, 2018, pp. 116–131.

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

[32] J. Yamagishi, C. Veaux, and K. MacDonald, "CSTR VCTK Corpus: English multi-speaker corpus for CSTR voice cloning toolkit," *The Centre for Speech Technology Research*, 2016.

[33] C. Yu, H. Lu, N. Hu, M. Yu, C. Weng, K. Xu, P. Liu, D. Tuo, S. Kang, G. Lei, D. Su, and D. Yu, "DurIAN: Duration informed attention network for multimodal synthesis," in *Proc. Interspeech*, 2020, pp. 2027–2031.

[34] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. ICML*, 2013.

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

[36] 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.

[37] 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.

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

[39] 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.

[40] Y. Ren, C. Hu, X. Tan, T. Qin, S. Zhao, Z. Zhao, and T.-Y. Liu, "FastSpeech 2: Fast and high-quality end-to-end text to speech," in *Proc. ICLR*, 2021.

[41] J. Donahue, S. Dieleman, M. Bińkowski, E. Elsen, and K. Simonyan, "End-to-end adversarial text-to-speech," in *Proc. ICLR*, 2021.

[42] J. Kim, J. Kong, and J. Son, "Conditional variational autoencoder with adversarial learning for end-to-end text-to-speech," in *Proc. ICML*, 2021, pp. 5530–5540.

[43] N. Chen, Y. Zhang, H. Zen, R. J. Weiss, M. Norouzi, N. Dehak, and W. Chan, "WaveGrad 2: Iterative refinement for text-to-speech synthesis," in *Proc. Interspeech*, 2021, pp. 3765–3769.

[44] X. Tan, J. Chen, H. Liu, J. Cong, C. Zhang, Y. Liu, X. Wang, Y. Leng, Y. Yi, L. He *et al.*, "NaturalSpeech: End-to-end text to speech synthesis with human-level quality," *arXiv preprint arXiv:2205.04421*, 2022.