A Fast and Lightweight Text-To-Speech Model with Spectrum and Waveform Alignment Algorithms

Kihyuk Jeong
DSP&AI Lab., Dept. of EE
Yonsei University, Seoul, Korea
khjeong@dsp.yonsei.ac.kr

Huu-Kim Nguyen
DSP&AI Lab., Dept. of EE
Yonsei University, Seoul, Korea
huukim136@dsp.yonsei.ac.kr

Hong-Goo Kang
DSP&AI Lab., Dept. of EE
Yonsei University, Seoul, Korea
hgkang@yonsei.ac.kr

***Abstract*—In this paper, we propose a fast and lightweight text-to-speech (TTS) model that generates high-quality speech even in CPU-only environments. By leveraging the front-end architecture of FastSpeech2, we adopt an effective generative adversarial network (GAN) framework for waveform synthesis, which enables training the proposed model in a fully end-to-end manner. Since the waveform generator consists of small-size convolutional networks, its inference speed is tremendously fast and the number of network parameters can be reduced by half compared to the FastSpeech2 model. However, the generated waveform segments are often not time-aligned with reference ones because of utilizing the predicted duration, which reduces the reliability of the discriminator module in the GAN framework. To solve the time mis-alignment problem, we propose a waveform alignment algorithm that synchronizes timing information between the reference and generated waveforms. In addition to the waveform aligning task, we include an auxiliary mel-spectrogram prediction task to further enhance perceptual quality. Since this task is only required for training, it does not increase the computational complexity during the inference stage. Objective and subjective experimental results show that the synthesized quality of the proposed model is comparable to that of conventional approaches.**

***Index Terms*—on-device TTS, waveform alignment, generative adversarial network (GAN)**

I. Introduction

The quality of speech generated from recent deep learning-based text-to-speech (TTS) models is indistinguishable from human voices, but this generally has come at the cost of TTS models becoming larger and more complex. In addition, their training processes and waveform generation speeds are inefficient and slow because the models take multiple sub-processing steps. For example, Tacotron2 [1] consists of two conversion steps: text to mel-spectrogram conversion by a front-end module, and mel-spectrogram to speech waveform generation by a neural vocoder. Since both front-end and neural vocoder [2] modules have an auto-regressive nature, the training and inference speeds are extremely slow. Moreover, the number of parameters is incredibly large—approximately 28 million only for the front-end module.

Several methods such as teacher-student training [3], [4], flow-based [5], and generative adversarial network (GAN)-based methods [6]–[9] were proposed to increase the generation speed in neural vocoding. To further increase the processing speed of the front-end module, CNN [10] and transformer-based models [11] were proposed; however, they still utilize an auto-regressive structure. FastSpeech [12] and FastSpeech2 [13] non-autoregressively estimate latent space embeddings of acoustic information from text inputs using a pre-trained feature prediction module, a so-called variance adaptor. Although the processing speeds of these methods are remarkably fast, they need to train neural vocoders with the estimated embeddings to generate speech waveforms.

Recently, the paradigm of TTS training has shifted from the aforementioned two-step approaches to fully end-to-end one-step approaches because of the success of waveform generation tasks with GANs [14]. In other words, by concatenating front-end and GAN-based waveform generation modules, it is possible to jointly train the two modules at once. However, although the training process becomes much simpler, it is not easy to obtain high-quality speech unless the training criterion is set appropriately. In addition, the computational complexity and the size of network vary by the structure of the generator and discriminator in the GAN-based waveform generation module.

In this paper, we propose a fast and compact end-to-end TTS model by concatenating the front-end feature extraction module from FastSpeech2 and GAN-style neural vocoders. As mentioned above, it is possible to directly generate speech waveforms in the GAN module using the latent space embeddings estimated in the front-end module. In FastSpeech2, phonetic information is estimated from a transformer and prosody information from a variance adaptor. To obtain high-quality speech, the embeddings should include both phonetic and prosody information appropriately.

However, output speech quality is degraded when the front-end and GAN-style neural vocoding modules are simply concatenated to build an end-to-end TTS system. The main causes of quality degradation are the prediction and discretization processes in the variance adaptor. As the predicted outputs of variance adaptors are approximated or represented by discretized values of pitch, energy, and duration, waveforms obtained by the generator of the GAN module are not well-aligned with reference waveforms. This is problematic in the discriminator module because the metrics used in the discriminator normally assume that reference and generated waveforms are time-synchronized. We propose an effective waveform pre-alignment process to solve the misalignment problem. Lastly, we additionally include an auxiliary mel-

spectrogram prediction task with the output embeddings of the front-end module. Due to the perceptually-motivated frequency characteristic, the quality of synthesized speech can be further improved. Similar to the problem occurring in the waveform generation process, predicted mel-spectrograms can also be affected by discretization, resulting in frame misalignment between target and predicted mel-spectrograms. We propose a locally matched spectral distance metric that considers delays to the predicted mel-spectrograms.

Our main contributions are as follows: 1) we propose a fast and lightweight on-device TTS framework by concatenating a non-autoregressive feature extraction module and a GAN-style waveform generation network module; 2) we propose an effective cross-correlation based segment alignment process to reliably measure the similarity between reference and generated waveforms in the discriminator network; 3) we further improve perceptual quality by applying perceptually motivated auxiliary feature prediction loss.

II. Related works

A. FastSpeech2

FastSpeech2 [13] is a transformer-based non-autoregressive TTS model. Since FastSpeech2 directly predicts prosody-related information such as duration, pitch, and energy using a jointly trained variance adaptor network, its generation speed is remarkably fast. Speech waveforms can be generated by either predicted mel-spectrograms with neural vocoding or a waveform decoding process. However, since waveform generation modules of the model require large networks, there is still room for reducing the size and complexity of the model. We use the front-end part of FastSpeech2 as our baseline for feature extraction, and introduce a light weight and fast GAN-style network to speed up the waveform generation process and reduce the model size.

B. GAN-based waveform generation

MelGAN [6] and HiFi-GAN [8] are typical examples of GAN-based neural vocoders. With a simple generator and a multi-scale discriminator (MSD), MelGAN greatly reduces model size and increases generation speed. Motivated by MelGAN, HiFi-GAN jointly utilizes a multi-period discriminator (MPD) as well as MSD. MPD faithfully captures the structural differences of equally spaced samples between the reference and generated waveforms, i.e. periodicity; thus, synthesized speech quality can be further enhanced. The architectures of the generator and discriminator in MelGAN and HiFi-GAN must be adjusted when their input features are not mel-spectrograms.

GAN-TTS is a GAN-style neural vocoder that generates speech waveforms by conditioning on linguistic and pitch information. It consists of two discriminators that use multi-frequency random windows. One of them is designed to consider linguistic conditions and the other is designed not to consider any specific information. These two discriminators use segments of speech as targets, and generate corresponding speech segments using a simple index-wise approach. To reliably measure discriminator scores, the waveform generator must synthesize waveform segments that are time-aligned with reference speech waveforms. End-to-end adversarial text-to-speech (ETAS) [14] is a fully end-to-end TTS model that also utilizes a GAN. The model is composed of an aligner and a decoder. The aligner produces low-frequency (200Hz) aligned embeddings using text/phoneme inputs. The decoder utilizes the GAN-TTS generator module and upsamples the obtained embeddings to obtain raw speech waveforms.

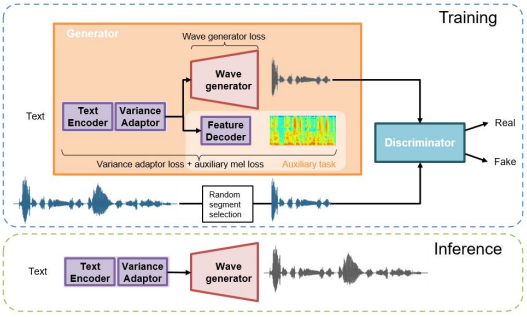

Fig. 1: Training and inference stages of our model

III. Proposed end-to-end TTS model

A. Overall block diagram

Fig. 1 illustrates the overall structure of our proposed model. The proposed model consists of a text encoder, a variance adaptor, a waveform generator with a discriminator, and an auxiliary mel-spectrogram feature prediction decoder. The text encoder estimates latent space phonetic embeddings, and then the phonetic embeddings are calibrated by the variance adaptor. The variance adaptor predicts prosody information such as pitch, duration, and energy terms, which is then added to the corresponding positions of phonetic embeddings. The text encoder and the variance adaptor are taken from the front-end module of FastSpeech2. The GAN-based waveform generator transforms the input embeddings into segments of speech waveforms using a convolutional neural network architecture. Our discriminator consists of two sub-discriminators: multi-scale discriminator (MSD) from MelGAN and multi-period discriminator (MPD) from HiFi-GAN. Those modules return the degree of differences of the pairs of reference and predicted speech segments. The auxiliary feature decoder predicts mel-spectrograms from the latent embeddings estimated in the front-end module, then computes an auxiliary mel-spectrogram prediction loss for back-propagation, which helps generate perceptually motivated latent embeddings. We used speech segments roughly 400ms in length (8,192 samples with a sampling frequency of 22.05 kHz) for training.

B. Discriminator scores in mis-aligned waveforms

The synthesized output quality of the proposed TTS model mentioned above is often degraded because of unreliable discriminator performance, especially when reference and generated speech segments are mis-aligned in time. In the proposed model, the duration and pitch interval of the phonetic

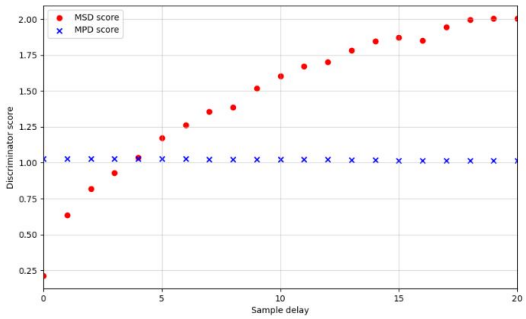

Fig. 2: Discriminator scores of MPD and MSD modules from only changing sample delays. The discriminator score is calculated by the sum of the waveform generator loss from the discriminator and feature matching loss, i.e. $\mathcal{L}_{Adv}(G;D)+\lambda_{fm}\mathcal{L}_{FM}(G;D)$. It is defined in Subsection III-D.

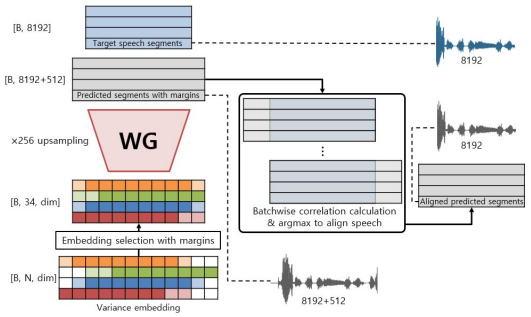

Fig. 3: Cross-correlation-based waveform alignment process. The dark colored parts are the desired target embedding blocks, and the light colored parts are margins needed for correlation matching.

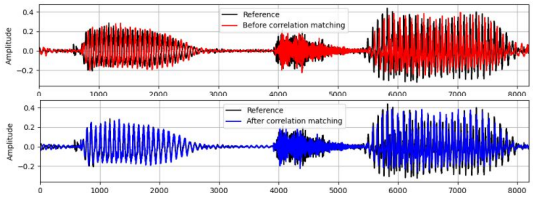

Fig. 4: Generated waveforms before and after the time alignment process. The upper one is before applying the alignment process, and the below one is after the alignment process.

information are determined by the variance adaptor module. However, we cannot perfectly synchronize the duration and pitch with the reference waveforms because of the prediction process and discretization step in the variance adaptor module. We found out that discriminator scores become inconsistent when the reference and generated waveforms are not synchronized. To validate our observation on the problem of waveform misalignment in the training process, we measured discriminator scores of the MSD and MPD modules using reference and time-shifted waveforms.

Fig. 2 shows the average discriminator scores of the MSD and MPD modules obtained by changing sample delays. The MSD module is immensely sensitive to misalignment because the average pooling process of the original and shifted signals changes signal characteristics. On the other hand, the MPD module is not affected by misalignment because it is designed to measure periodicity, while the periodicity of original and shifted ones do not change much from sample delays.

C. Waveform alignment process

From the experiments mentioned above, we realize that it is crucial to time-align the generated waveforms with the reference ones to effectively perform adversarial training. To solve the problem caused by mis-alignment, we propose a batch-wise cross-correlation method that finds the best matched speech segments from the predicted speech waveforms.

The overall process is illustrated in Fig.3. The process begins with the selection of embedding blocks with margins. N is the maximum length of an embedding in a batch, and the length of the core embedding block corresponding to the target waveform is set to 32. By including one additional block before and after the core embedding block, we obtain a batch with a total length of 34. The waveform generator (WG) generates speech samples that are up-sampled by 256 times; thus, the length of the generated waveform segment is 8704. By applying batch-wise cross-correlation, we obtain predicted segments that are well-matched with target waveforms. Note that we only need to calculate cross-correlation for the lag of 2×256 indices; thus, the cross-correlation processing does not significantly increase training time. Fig. 4 shows an example of generated waveforms before and after the time alignment process. Compared to the upper one that does not apply the alignment process, the time-aligned waveform (below figure) tends to follow the shape of reference waveform more alike especially at both ends.

D. Training objective

The overall training losses of our model are as follows:

$$\mathcal{L}_G = \mathcal{L}_{Adv}(G;D) + \lambda_{fm}\mathcal{L}_{FM}(G;D) + \lambda_{mel}\mathcal{L}_{Mel}(G) + \lambda_{tts}\mathcal{L}_{TTS}(G) \quad (1)$$

$$\mathcal{L}_D = \mathcal{L}_{Adv}(D;G) \quad (2)$$

$$\mathcal{L}_{TTS}(x) = \mathcal{L}_{Var}(x) + \mathcal{L}_{aux}(x) \quad (3)$$

$$\mathcal{L}_{Var}(x) = \mathbb{E}[||(d,p,e) - (d,p,e)'||] \quad (4)$$

$$\mathcal{L}_{aux}(x) = \mathbb{E}[||mel - mel'||_1] + \frac{1}{K}\sum_{k=1}^{K}[||mel_k - mel'_k||_1] \quad (5)$$

, where $\lambda_{fm}=2$, $\lambda_{mel}=45$ and $\lambda_{tts}=1$.

Due to the nature of GAN training, the training criterion of the proposed method consists of generator loss $\mathcal{L}_G$ (1) and discriminator loss $\mathcal{L}_D$ (2), where $\mathcal{L}_{Adv}(G;D)$, $\mathcal{L}_{FM}(G;D)$, $\mathcal{L}_{Mel}(G)$, and $\mathcal{L}_{TTS}(G)$ denote waveform generator loss, feature matching loss, mel-spectrogram loss, and TTS loss, respectively. The waveform generator and discriminator use LS-GAN [15] loss to avoid a vanishing gradient problem. The feature matching loss measures the L1 distance between

intermediate embeddings between reference and generated speech taken from the discriminator. The mel-spectrogram loss that measures the spectral distance between reference and generated speech is also used to improve training efficiency and perceptual quality.

The TTS loss (3) is composed of variance adaptor loss $\mathcal{L}_{Var}$ (4) and feature prediction loss $\mathcal{L}_{aux}$ (5): where x, d, p, e, K are input text, duration, pitch, energy and number of chunks, respectively. The variance adaptor loss is measured by the L2 distance between target and predicted pitch, duration and energy. In addition, we use L1 loss between the target and predicted mel-spectrograms as an auxiliary task. We added the chunk-wise aligned spectral distance to compensate for delays in predicted features. The method uses chunks of target and predicted features that have a length of 8 and executes an alignment process that is similar to the one in the waveform alignment process. The spectral matching loss was included in the criterion when the training step reaches 300k. Otherwise, the overall performance degrades because the untrained feature prediction module generates distorted outputs at the early training stage of the model. As mel-spectrograms use a mel-scale that is highly related to human perception, we expect that the model emphasizes perceptually more important frequency bands, resulting in better performance.

Although $\mathcal{L}_{Mel}(G)$ and $\mathcal{L}_{aux}(x)$ both measure mel-spectrogram distances, they do so in different ways. When we calculate $\mathcal{L}_{Mel}(G)$, mel-spectrograms are computed with waveform segments obtained from the output of the generator, which focus on the local part of training. To calculate $\mathcal{L}_{aux}(x)$, the predicted mel-spectrogram is taken from the auxiliary feature prediction decoder that corresponds to the whole sentence, influencing the global part of model training.

IV. Experimental results

A. Data and model setup

In our experiments, we used the LJSpeech dataset, which is composed of around 24 hours of audio clips recorded by single female speaker. We used a 22.05kHz sampling rate and trimmed silence using alignment information earned by MFA tools. The features such as energy, pitch, duration and mel-spectrograms were extracted with the analysis window and fast Fourier transform length of 1024 and hop length of 256. In our model, the encoder and feature decoder are formed using multiple layers of a feedforward transformer block that consists of multi-head attention, 2-layer 1D convolution block and batch normalization with residual connections. The variance adaptor is composed of duration, pitch, and energy predictors. Each predictor consists of two 1D convolutional layers with layer normalization and dropout in between, followed by a linear layer to predict variance factors (duration, pitch, energy). We follow the architecture of HiFi-GAN for the waveform generator and discriminator blocks in our model. We trained the model for 800k steps with a batch size of 16 on a single Nvidia GTX-1080 Ti GPU. To generate the samples, we only used a CPU-only processor, Intel(R) Core(TM) i7-7700 CPU @ 3.60GHz, with a single thread.

TABLE I: Model size and generation speed. The input texts consist of roughly 200 characters each.

Model type	# of parameters	Generation speed(RTF)
Tacotron2 + HifiGAN	29.4M	0.718
FastSpeech2 + HifiGAN	28.4M	0.292
Proposed model	14.4M	0.199

B. Model size and generation speed

In this section, we compare model size and generation speed. For comparison, we also include results from pre-trained Tacotron2 and FastSpeech2 with the pre-trained HiFi-GAN vocoder. Note that the spectral matching (SM), waveform alignment process (WAP) and mel feature prediction (MFP) processes were only used for training; thus, there is no impact on generation speed and model size at inference time. The model size was measured in terms of the number of parameters. As shown in Table I, Tacotron2 and FastSpeech2 require more than 28M parameters in total. On the other hand, our model requires only 14.4M, about half the number of parameters compared to the two conventional models.

The generation speed was measured using the real-time factor (RTF) scale. The RTF is defined as the time spent (in seconds) for the model to generate one second of speech; thus, the higher the value, the lower the generation speed. As our main objective is to develop a method for on-device TTS, we only measured generation speed in a CPU environment by limiting the process to a single thread. The pieces of input text that were used in this experiment had roughly 200 characters each, which is challenging. As shown in Table I, the Tacotron2 model had the highest RTF, which was about 2.5 times slower than the FastSpeech2 model. This is because it has an auto-regressive structure, unlike the other models. On the other hand, our proposed model is about 1.5 times faster than the FastSpeech2 model because we directly synthesize the speech without predicting intermediate features in the inference stage.

C. Mean opinion score

To evaluate the perceptual quality of the models, we conducted a MOS test. Sixteen participants were asked to give scores from 1 to 5, where a higher score indicated better performance. We also include ablation tests to validate our assumptions. To compare the performance with other state-of-the-art TTS models, we included Tacotron2 and FastSpeech2 with pre-trained GAN-based vocoders in the MOS test. As shown in Table II, Tacotron2 had the highest score because of using an auto-regressive structure that reflects contextual flow better. The FastSpeech2 model received an average score of 3.89, indicating that the model generates audio samples of reasonable quality. On the other hand, our proposed model had an average score of 3.93, slightly outperforming the FastSpeech2 model. Ablation test results show that all of the proposed approaches (spectral matching (SM), mel-feature prediction (MFP), and waveform alignment process (WAP)) are helpful for improving perceptual quality. The effectiveness of SM and MFP were higher than that of WAP.

TABLE II: MOS results with 95% confidence intervals (SM: spectral matching, MFP: mel-feature prediction, WAP: waveform alignment process)

Number	Model description	MOS
1	Tacotron2 + HifiGAN	4.16 ± 0.09
2	FastSpeech2 + HifiGAN	3.89 ± 0.10
3	Proposed model (PM)	3.93 ± 0.09
4	PM without SM	3.78 ± 0.09
5	PM without SM and MFP	3.55 ± 0.10
6	PM without SM and WAP	3.62 ± 0.11
7	PM without SM, WAP and MFP	3.42 ± 0.10

TABLE III: Objective measurements

Number	Model type	MCD (dB)	F0 RMSE (Hz)
2	Baseline model	6.44	70.38
4	Proposed model	6.10	53.50
7	Vanilla model	6.49	54.78

D. Objective measurement

We also included objective measurement scores [16]: mel-cepstral distance (MCD) and F0 root mean square error (RMSE). Since these metrics are difficult to use in the inference stage because the predicted prosody features are different from the ground truth, we used the samples in the validation stage instead. In this stage, rather than predicting variance embeddings from input text, it uses reference duration, pitch and energy. Thus, the prosodies of the generated waveforms are similar to the ground truth. As shown in Table III, all of the tested models received similar MCD scores, but our proposed model had the best score. Baseline model denotes FastSpeech2, and Vanilla model denotes model number 7 in Table II. Our proposed model also had the best score for the F0 RMSE metric. We believe that this is because the Baseline model accumulates distortions by taking a two step approach in the generation process.

V. Conclusion

In this paper, we have proposed an end-to-end on-device text-to-speech (TTS) system framework that significantly reduces model size and complexity by directly concatenating a GAN-style waveform generator to the front-end module. To overcome the time-domain mis-alignment problem of the proposed system, we proposed alignment processes for the generated speech waveform and mel-spectrograms. With these alignment processes, the discriminator module and auxiliary feature prediction loss are able to fairly compare the similarities between target and generated parameters, resulting in improved performance. For future work, there still remain challenges in model size, complexity, and quality. As most parts of our model are based on CNNs, we expect that more effective model compression methods [17] will be needed to further reduce the size and complexity. To further enhance synthesized quality, it is necessary to re-design the front-end module to effectively estimate latent embeddings for representing phonetic and prosody information. Introducing other types of auxiliary feature prediction modules may be another feasible direction.

VI. Acknowledgements

The work was supported by Clova Voice, NAVER Corp., Seongnam, Korea.

References

[1] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerrv-Ryan, *et al.*, "Natural tts synthesis by conditioning wavenet on mel spectrogram predictions," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4779–4783, IEEE, 2018.

[2] A. v. d. 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.

[3] W. Ping, K. Peng, and J. Chen, "Clarinet: Parallel wave generation in end-to-end text-to-speech," in *7th International Conference on Learning Representations, ICLR 2019*.

[4] A. Oord, Y. Li, I. Babuschkin, K. Simonyan, O. Vinyals, K. Kavukcuoglu, G. Driessche, E. Lockhart, L. Cobo, F. Stimberg, *et al.*, "Parallel wavenet: Fast high-fidelity speech synthesis," in *International conference on machine learning*, pp. 3918–3926, PMLR, 2018.

[5] R. Prenger, R. Valle, and B. Catanzaro, "Waveglow: A flow-based generative network for speech synthesis," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3617–3621, IEEE, 2019.

[6] K. Kumar, R. Kumar, T. de Boissiere, L. Gestin, W. Z. Teoh, J. Sotelo, A. de Brébisson, Y. Bengio, and A. C. Courville, "Melgan: Generative adversarial networks for conditional waveform synthesis," in *Advances in Neural Information Processing Systems* (H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, eds.), vol. 32, Curran Associates, Inc., 2019.

[7] 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 *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6199–6203, IEEE, 2020.

[8] J. Kong, J. Kim, and J. Bae, "Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis," in *Advances in Neural Information Processing Systems* (H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, eds.), vol. 33, pp. 17022–17033, Curran Associates, Inc., 2020.

[9] M. Binkowski, J. Donahue, S. Dieleman, A. Clark, E. Elsen, N. Casagrande, L. C. Cobo, and K. Simonyan, "High fidelity speech synthesis with adversarial networks," in *8th International Conference on Learning Representations, ICLR 2020*.

[10] H. Tachibana, K. Uenoyama, and S. Aihara, "Efficiently trainable text-to-speech system based on deep convolutional networks with guided attention," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4784–4788, IEEE, 2018.

[11] N. Li, S. Liu, Y. Liu, S. Zhao, and M. Liu, "Neural speech synthesis with transformer network," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 6706–6713, 2019.

[12] Y. Ren, Y. Ruan, X. Tan, T. Qin, S. Zhao, Z. Zhao, and T.-Y. Liu, "Fastspeech: Fast, robust and controllable text to speech," *arXiv preprint arXiv:1905.09263*, 2019.

[13] Y. Ren, C. Hu, T. Qin, S. Zhao, Z. Zhao, and T.-Y. Liu, "Fastspeech 2: Fast and high-quality end-to-end text-to-speech," *arXiv preprint arXiv:2006.04558*, 2020.

[14] J. Donahue, S. Dieleman, M. Binkowski, E. Elsen, and K. Simonyan, "End-to-end adversarial text-to-speech," in *9th International Conference on Learning Representations, ICLR 2021*.

[15] X. Mao, Q. Li, H. Xie, R. Y. Lau, Z. Wang, and S. Paul Smolley, "Least squares generative adversarial networks," in *Proceedings of the IEEE international conference on computer vision*, pp. 2794–2802, 2017.

[16] T. Hayashi, A. Tamamori, K. Kobayashi, K. Takeda, and T. Toda, "An investigation of multi-speaker training for wavenet vocoder," in *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pp. 712–718, IEEE, 2017.

[17] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.