Interspeech 2022
18-22 September 2022, Incheon, Korea

One-Shot Speaker Adaptation Based on Initialization by Generative Adversarial Networks for TTS

*Jaeuk Lee and Joon-Hyuk Chang**

Department of Electronic Engineering Hanyang University Seoul, Republic of Korea
pier314@hanyang.ac.kr, jchang@hanyang.ac.kr

Abstract

Speaker adaptation for personalizing text-to-speech (TTS) has become increasingly important. Herein, we propose a novel adaptation using a few seconds of data obtained from an unseen speaker. We first use a speaker embedding lookup table to train a multi-speaker TTS model, wherein each speaker embedding in the lookup table contains information representing a speaker's timbre. We propose an initial embedding predictor that extracts initial embedding suitable for the adaptation of unseen speakers. We use trained speaker embeddings to train the initial embedding predictor. Further, adversarial training is applied to improve the performance. After adversarial training, the initial embedding predictor infers the unseen speaker's initial embedding, and it is fine-tuned. As the initial embedding contains timbre information of the unseen speaker, adaptation is achieved faster and with less data than with conventional methods. We validate the performance with a mean opinion score (MOS) and demonstrate that adaptation is feasible with only 5 s of data.

Index Terms: speaker adaptation, voice cloning, multi-speaker

1. Introduction

Several studies have been conducted on multi-speaker text-to-speech (TTS) systems that synthesize the speech of multiple speakers using a TTS model. Herein, the speaker embeddings for each speaker are employed for the speaker condition of the TTS model. DeepVoice2 [1] used trainable speaker embeddings based on a lookup table for the first time. The speaker embedding table is trained with the TTS model to generate suitable speaker embeddings to express each speaker through back-propagation from the TTS model. There exist two primary method for generating a speaker embedding: one uses a speaker encoder [2, 3] to extract the speaker embedding from a reference mel spectrogram, and the other uses the speaker embedding table as mentioned above. GST Tacotron [4], style embedding based on attention mechanisms to mimic a speaker's style. In [5], a multi-speaker TTS model with a speaker embedding table performs better than a speaker encoder. There are two methods to adapt an unseen speaker to a trained multi-speaker TTS model with a speaker embedding table or speaker encoder.

- **Zero-shot voice cloning:** The speaker encoder estimates a speaker embedding from an unseen speaker's reference mel spectrogram. Thereafter, the trained multi-speaker TTS model uses the speaker embedding to synthesize a speech possessing an unseen speaker's timbre without fine-tuning.
- **Speaker adaptation:** Generally, multi-speaker TTS model based on a speaker embedding table is fine-tuned with an unseen speaker data (audio and text).

* Corresponding author.

As speaker adaptation involves fine-tuning, it produces considerably better performance than zero-shot voice cloning. For fine-tuning in speaker adaptation [6, 7], there exist several approaches to update the parameters of the entire TTS model [8], part of the model [9], or only the speaker embedding. In [8], effective fine-tuning of the entire model, including the speaker embedding table, was demonstrated. In previous studies [10, 11], the unseen speaker embedding was randomly initialized before fine-tuning the entire model with unseen speaker data. However, it is difficult to expect good performance with limited data within 10 s after random initialization. In [8], after random initialization, only the speaker embedding is fine-tuned before the entire model is fine-tuned again. Early stopping, which limits the number of training steps, was employed to prevent overfitting with limited data. In [12], an unseen speaker embedding was initialized with a speaker embedding possessing a similar timbre with the unseen speaker. As speakers with similar timbres have similar speaker embedding, the results produced are better than those produced with random initialization. Thus, it can be concluded that the limited the amount of data used to fine-tune the TTS model, the more critical the initialization of speaker embedding is.

In this paper, we propose a generative adversarial network (GAN) [13]-based initial embedding predictor that predicts the initial embedding of an unseen speaker. First, we apply the speaker embedding table to Tacotron2 [14] and train the model with multi-speaker datasets. Thereafter, we should initialize an unseen speaker embedding to adapt the unseen speaker to the multi-speaker TTS model. In previous studies, a speaker embedding with a timbre similar to that of the unseen speaker was used as the initial embedding for the unseen speaker, or the unseen speaker embedding was randomly initialized. However, our initial embedding predictor generates an initial embedding that effectively represents the timbre of an unseen speaker. The initial embedding predictor requires two inputs: trained speaker embeddings with the timbre similar to that of the unseen speaker, and a d-vector extracted from audio of the unseen speaker. In Sections 3 and 4, we demonstrate that initial embeddings generated by the initial embedding predictor include timbre information for the unseen speaker and that our method is more effective than other initialization methods.

2. Proposed Method

We design a multi-speaker TTS model based on Tacotron2 with the speaker embedding table consisting of several speaker embeddings, and each speaker embedding is extracted corresponding to an entered speaker ID as follows:

$$e_m = E_O(m) \cdot T, \tag{1}$$

where m and E_O denote the $m-$th speaker ID and one-hot encoding, respectively. T is a speaker embedding table consisting

Copyright © 2022 ISCA

10.21437/Interspeech.2022-934

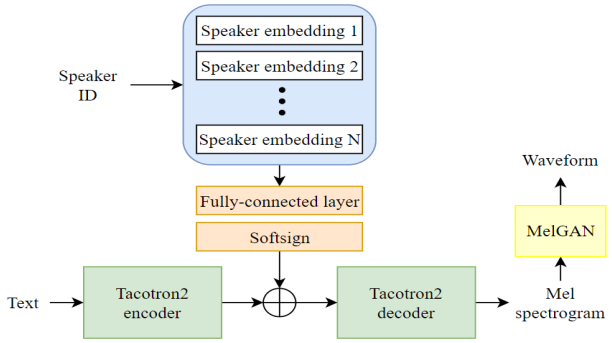


Figure 1: *Structure of speaker embedding table-based multi-speaker Tacotron2.*

of N speaker embeddings. The dimension of a speaker embedding is d_e thus, T has dimension of $N \times d_e$. The extracted speaker embedding is added to the encoder output, as shown in Fig. 1. Here, softsign is the activation function, which can be described as follows:

$$\text{softsign}(x) = \frac{x}{1+|x|}. \tag{2}$$

We train the multi-speaker Tacotron2 with N speakers and obtain N speaker embeddings. Additionally, MelGAN [15] is used as a vocoder to convert the mel spectrogram into a waveform.

2.1. Initial Embedding Predictor

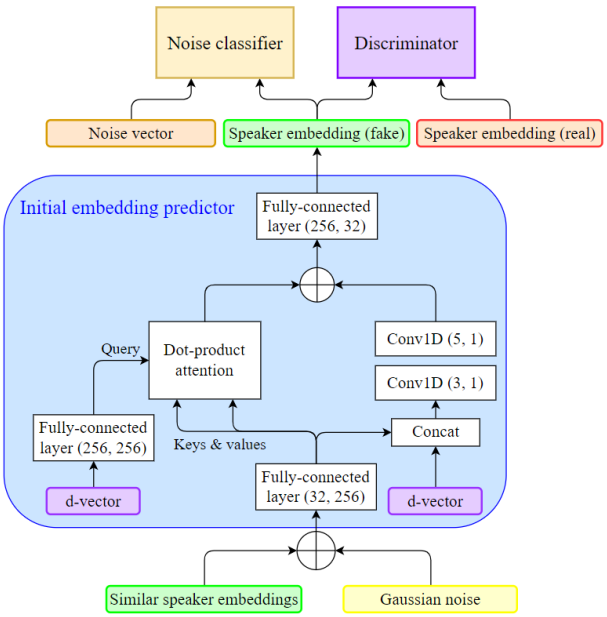


Figure 2: *Initial embedding predictor with adversarial training and noise classifier. 'Conv1D (n, m)' means the kernel size and stride size in 1D convolution are n and m. 'Fully-connected layer (l, k)' means the dimension of input and output is l and k, respectively.*

Through multi-speaker training, we obtain N speaker embeddings including the timbre information for each speaker. These speaker embeddings are utilized to train the initial embedding predictor. Of the N speaker embeddings, we designate a target speaker embedding that the initial embedding predictor should generate. Thereafter, we select K speaker embeddings with a timbre similar to that of the target speaker. We evaluate the timbre similarity using the cosine similarity of the d-vector extracted from the speaker verification model [16] as follows:

$$\text{cosine similarity}(d_i, d_{target}) = \frac{d_i \cdot d_{target}}{\|d_i\|\|d_{target}\|}, \tag{3}$$

where d_i represents the d-vector of the $i-$th speaker among trained speakers, and d_{target} is the d-vector of the target speaker. Also, $\|\cdot\|$ denotes the $l2$ norm. We select K speakers in the order of high cosine similarity with the target speaker. Consequently, the K speaker embeddings become 'similar speaker embeddings' that served as the input to the initial embedding predictor depicted in Fig. 2.

2.1.1. Generator

We use an attention mechanism and convolutional neural networks (CNNs) to convert the K speaker embeddings into a single latent vector, as shown in Fig. 2. The initial embedding predictor structure combines two latent vectors extracted by both CNNs and attention mechanism, respectively. It produces better results than those produced using either of them.

2.1.2. Loss function

We apply adversarial training to improve the performance of the initial embedding predictor. The discriminator distinguishes the real speaker embedding generated by multi-speaker training from fake speaker embedding generated by the initial embedding predictor. The discriminator consists of 4 fully-connected layers and sigmoid function. Adversarial training with the discriminator causes the distribution of fake speaker embedding to resemble that of real speaker embeddings. The loss function of the discriminator [13] can be expressed as follows:

$$L_D = \mathbb{E}_x[-\log(D(x))] + \mathbb{E}_{I,z}[-\log(1-D(G(I,z)))], \tag{4}$$

where x is the real speaker embedding, and I and z represent similar speaker embeddings and Gaussian noise, respectively. With G as the initial embedding predictor acting as a generator, we train the discriminator using L_D. We additionally use a noise classifier to classify the noise vector and fake speaker embedding. The noise classifier consists of 4 fully-connected layers and sigmoid function. The auxiliary classifier improved the performance of GANs as in [17]. Therefore, we exploit the noise classifier so that the generator yields a speaker embedding that is distinguishable from the noise. We generate a noise vector sampled from Gaussian or uniform distribution. The noise classifier is trained to distinguish the speaker embeddings from the noise vector. The loss function is described as follows:

$$L_N = \mathbb{E}_{v_N}[-\log(1-C_N(v_N))] + \mathbb{E}_{I,z}[-\log(C_N(G(I,z)))], \tag{5}$$

where v_N and C_N are the noise vector and noise classifier, respectively, unlike L_D, back-propagation from L_N continues to the generator and is trained jointly. To reduce the loss of the noise classifier, the generator produces a speaker embedding that is far from the Gaussian and uniform distribution. Consequently, the generated speaker embedding moves closer to real distribution. The loss function of the generator consists of the

following 3 terms:

$$L_G = \mathbb{E}_{I,z}[-\log(D(G(I,z)))] + \lambda * L_r + L_N, \quad (6)$$

where the first term is prepared for adversarial training with the discriminator, and L_r is the reconstruction loss of the mean squared error between real and fake speaker embedding. In addition, λ is a hyperparameter set to 2.

2.2. Unseen Speaker Adaptation

After training the initial embedding predictor, the predictor infers an initial embedding of an unseen speaker. We obtain the d-vector from a short utterance from the unseen speaker. Thereafter, we compute the cosine similarity between the trained speakers and the unseen speaker, selecting K speakers with a high similarity to the unseen speaker. Consequently, K speaker embeddings become the input for the initial embedding predictor to generate an initial embedding for the unseen speaker. The initial embedding is fine-tuned with unseen speaker data consisting of the audio from which the d-vector was extracted and transcript corresponding to the audio. The initial embedding already includes the information representing the timbre of the unseen speaker, so fine-tuning is completed quickly.

3. Experiments

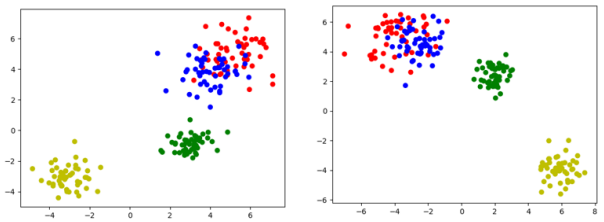
In this section, we present an analysis of the proposed method from several perspectives. We first validated the positive effects of noise classifier and adversarial training. Additionally, we also demonstrate that our method is capable of adapting to the unseen speaker faster and with less data than previous initialization methods involving limited data.

3.1. Experimental Setup

We used the VCTK [18] and LibriTTS [19] datasets for this study. Our training set consisted of 108 speakers from the VCTK dataset and 792 speakers from the LibriTTS dataset. The audio was downsampled to 22050 Hz, and the mel spectrogram was extracted with a hop size of 256 and window size of 1024. The multi-speaker TTS model was trained using 900 speakers, thereby producing 900 trained speaker embeddings. To train the initial embedding predictor, 850 and 50 speaker embeddings were considered as the training set and validation set, respectively. The dimension of the speaker embedding was 32. The learning rates for Tacotron2 and MelGAN were $1e^{-3}$ and $1e^{-4}$, respectively, for multi-speaker training, and we used Adam optimizer [20]. The learning rate for the initial embedding predictor was $2e^{-4}$, and Adam optimizer was used to train it. When we fine-tuned Tacotron2 and the initial embedding for speaker adaptation, we set the learning rate to $1e^{-4}$. After several experiments, the best performance was observed when 7 similar speaker embeddings were selected and used as the input to the initial embedding predictor so we set K to 7.

3.2. Effect of Noise Classifier

We verified the effect of the noise classifier using the validation set of 50 speaker embeddings. As shown in Fig. 3, the noise classifier moves the generated speaker embeddings away from noise vector sampled from Gaussian and uniform distributions. Accordingly, the generated speaker embedding was closer to the target. Although the addition of the noise classifier did not significantly improve the performance, as shown in Table 1, this small difference could be significant in zero-shot voice cloning.



(a) (b)

Figure 3: *tSNE for visualizing distribution of generated speaker embeddings based on the usage of noise classifier. (b) and (a) show result of using and not using noise classifier, respectively. Each color represents as follows: yellow = vectors sampled from Gaussian distribution, green = vectors sampled from uniform distribution [-0.1, 0.1], blue = generated speaker embeddings and red = real speaker embeddings.*

3.3. Adversarial Training

Table 1: *Result of $l1$-distance between target speaker embedding and initialized embedding at validation set. AT is adversarial training and d_e is dimension of speaker embedding. NC is noise classifier. RS denotes that the input of the initial embedding predictor in training and inference stages is speaker embeddings not selected based on cosine similarity but randomly selected.*

Initialization method	**l1-distance/d_e**
Uniform distribution (-0.1, 0.1)	0.1082
Gaussian distribution ($\mu = 0$, $\sigma = 1$)	0.2231
Similar speaker embedding	0.0671
Ours (RS)	0.0549
Ours (w/o AT)	0.0227
Ours (w/o NC)	0.0161
Ours	0.0142

This section describes the benefits of adversarial training of the initial embedding predictor. With adversarial training, $l1$-distance between the generated and target speaker embeddings is smaller than without it, as shown in Table 1. We also observed that the average $l1$-distance and standard deviation decreased. With adversarial training, the standard deviation was $2.31e^{-3}$, whereas without it, standard deviation was $3.43e^{-3}$. Our method yielded better performance than that yielded by an initialization using a speaker embedding with a timbre similar to that of the target speaker. Adversarial training produced a distribution of generated speaker embeddings closer to the target distribution. We observed that not using similar speaker embeddings as the input to the initial embedding predictor resulted in significant performance degradation.

3.4. Fast Adaptation

We randomly selected 2 utterances from an unseen speaker that were not used in multi-speaker training. The utterance of the unseen speaker used for validation is shown in Fig. 4 (a). We fine-tuned the TTS model and the initial embedding generated from the initial embedding predictor using an utterance different from the validation utterance. The resulting mel spectrogram includes a harmonic similar to that of the unseen speaker, as shown in Fig. 4 (c). When fine-tuning a similar speaker embed-

ding, adaptation occurs later, and the harmonic is less similar to that of the unseen speaker, as shown in Fig. 4. (b). Fine-tuning using a similar speaker embedding required more than 50 epochs. Herein, we exhibit only one case, but our method also has a faster adaptation speed than similar speaker embedding even in using other speakers.

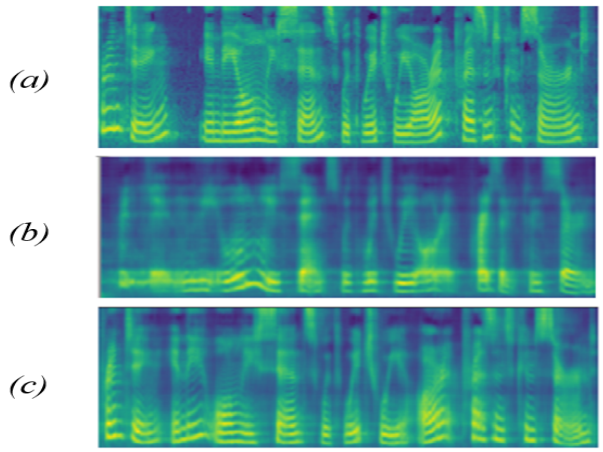


Figure 4: *Mel spectrogram: (a) ground truth of an unseen speaker not included in multi-speaker training set, training with 30 epochs started from (b) similar speaker embedding and (c) initial embedding generated from the initial embedding predictor.*

4. Evaluation

This section compares our study with previous studies. For this purpose, we conducted a crowd-sourced mean opinion score (MOS) test to evaluate the naturalness and similarity of the synthesized speech. We prepared a test set of 30 speakers that were not included in the training or validation sets. For the evaluation of naturalness MOS, we generated an average of 8 utterances for each speaker so the total number of utterances was 240. The ground truth denoted the real recorded audio. Thereafter, we evaluated the timbre similarity by pairing real audio with generated audio for the same speaker. For each speaker in the test set, we randomly selected one 5 s sample for adaptation and zero-shot voice cloning.

First, we conducted zero-shot voice cloning, thereby synthesizing speech without fine-tuning. The zero-shot voice cloning performance indicated that the unseen speaker embedding generated by the initial embedding predictor contained timbre information. Our study was compared with [5], which is a well-known study on zero-shot voice cloning. The study in [5] concatenated the d-vector extracted from the speaker verification model with the encoder output of the Tacotron2. Our method achieved better performance in terms of the naturalness and similarity MOS as shown in Tables 2 and 3.

Next, fine-tuning for speaker adaptation was performed using the same utterance that was used for zero-shot voice cloning. Representative initialization methods for speaker adaptation are described as follows: The study [12] exploits a trained speaker embedding, with a timbre that was most similar to the unseen speaker, as the unseen speaker's initial embedding. The above method describes MSE (most similar speaker embedding) in Tables 2 and 3. Another study [8] randomly initialized the unseen speaker embedding. The speaker embedding was fine-tuned before the entire TTS model (including the speaker embedding) was fine-tuned again. Our study achieves better adaptation results than these previous studies in terms of naturalness and similarity, as shown in Tables 2 and 3.

Finally, we observed changes in the performance by applying our initialization to AdaSpeech [21]. It should be noted that AdaSpeech is not related to initialization. However, scales and bias were obtained from the speaker embedding, and each speaker performed layer normalization differently. Fine-tuning of AdaSpeech for adaptation was conducted as described in [21]. We observed that our initialization improved the performance of AdaSpeech in terms of the naturalness and similarity.

Table 2: *Result of naturalness MOS with 95% confidence interval of each method. MSE is most similar speaker embedding used to unseen speaker's initial embedding.*

Ground Truth		4.46±0.06
Method	**Zero-shot**	**Adaptation**
[5]	3.27±0.09	-
AdaSpeech	-	3.43±0.11
[8] (w/ Tacotron2)	-	3.51±0.10
MSE (w/ Tacotron2)	-	3.56±0.08
Ours (w/ Tacotron2)	**3.41±0.08**	**3.62±0.07**
Ours (w/ AdaSpeech)	3.36±0.09	3.60±0.11

Table 3: *Result of similarity MOS with 95% confidence interval of each method.*

Ground Truth		4.37±0.08
Method	**Zero-shot**	**Adaptation**
[5]	2.91±0.12	-
AdaSpeech	-	3.32±0.11
[8] (w/ Tacotron2)	-	3.55±0.09
MSE (w/ Tacotron2)	-	3.64±0.10
Ours (w/ Tacotron 2)	3.09±0.11	**3.78±0.10**
Ours (w/ AdaSpeech)	**3.15±0.10**	3.57±0.09

5. Conclusion

Herein, we proposed a novel initialization method for speaker adaptation. We designed an initial embedding predictor and employed adversarial training to obtain an initial embedding. We demonstrated that our GAN-based initial embedding predictor generated speaker embeddings better represented unseen speakers than speaker embeddings in previous studies. Moreover, we demonstrated that zero-shot voice cloning was possible using an initial embedding predictor, and it exhibited better performance than that of previous study based on use of only d-vector.

6. Acknowledgements

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No.2020-0-01373, Artificial Intelligence Graduate School Program(Hanyang University)).

7. References

[1] S. Arik, G. Diamos, A. Gibiansky, J. Miller, K. Peng, W. Ping, J. Raiman and Y. Zhou, "Deep voice 2: multi-speaker neural text-to-speech," in *Proc. Advances in Neural Information Processing Systems* (NeurIPS), 2017, pp. 2966–2974.

[2] S. O. Arik, J. Chen, K. Peng, W. Ping and Y. Zhou, "Neural voice cloning with a few samples," in *Proc. Advances in Neural Information Processing Systems* (NeurIPS), 2018, pp. 10040–10050.

[3] E. Cooper, CI. Lai, Y. Yasuda, F. Fang, X. Wang, N. Chen and J. Yamagishi, "Zero-shot multi-speaker text-to-speech with state-of-the-art neural speaker embeddings," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing* (ICASSP), 2020, pp. 6184-6188.

[4] Y. Wang, D. Stanton, Y. Zhang, R. S.-Ryan, E. Battenberg, J. Shor, Y. Xiao, F. Ren, Y. Jia and R. A. Saurous, "Style tokens: unsupervised style modeling, control and transfer in end-to-end speech synthesis," in *Proc. Advances in Neural Information Processing Systems* (NeurIPS), 2018, pp. 5167-5176.

[5] Y. Jia, Y. Zhang, R. J. Weiss, Q. Wang, J. Shen, F. Ren, Z. Chen, P. Nguyen, R. Pang, I. L. Moreno and Y. Wu, "Transfer learning from speaker verification to multi-speaker text-to-speech synthesis," in *Proc. Advances in Neural Information Processing Systems* (NeurIPS), 2018, pp. 4485–4495.

[6] Y. Yan, X. Tan, B. Li, T. Qin, S. Zhao, Y. Shen and T.-Y. Liu, "Adaspeech 2: adaptive text to speech with untranscribed data," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing* (ICASSP), 2021, pp. 6613-6617

[7] Y. Yan, X. Tan, B. Li, G. Zhang, T. Qin, S. Zhao, Y. Shen, W.-Q. Zhang and T.-Y. Liu, "Adaspeech 3: adaptive text to speech for spontaneous style," in *Proc. INTERSPEECH*, 2021.

[8] Y. Chen, Y. Assael, B. Shillingford, D. Budden, S. Reed, H. Zen, Q. Wang, L. C. Cobo, A. Trask, B. Laurie, C. Gulcehre, Aä. van den Oord, O. Vinyals and N. de Freitas, "Sample efficient adaptive text-to-speech," in *Proc. International Conference on Learning Representations* (ICLR), 2019.

[9] G. Sun, Y. Zhang, R. J. Weiss, Y. Cao, H. Zen, A. Rosenberg, B. Ramabhadran and Y. Wu, "Generating diverse and natural text-to-speech samples using a quantized fine-grained VAE and auto-regressive prosody prior," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing* (ICASSP), 2020, pp. 6699-6703.

[10] M. Blaauw, J. Bonada and R. Daido, "Data efficient voice cloning for neural singing system," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing* (ICASSP), 2019, pp. 6840-6844.

[11] Y. Xiao, L. He, H. Ming and F. K. Soong, "Improving prosody with linguistic and bert derived Features in multi-speaker based Mandarin Chinese neural TTS," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing* (ICASSP), 2020, pp. 6704-6708.

[12] J. Cong, S. Yang, L. Xie, G. Yu and G. Wan, "Data efficient voice cloning from noisy samples with domain adversarial training," in *Proc. INTERSPEECH*, 2020.

[13] I. J. Goodfellow, J. P.-Abadie, M. Mirza, B. Xu, D. W.-Farley, S. Ozair, A. Courville and Y. Bengio, "Generative adversarial networks," *in Proc. Advances in Neural Information Processing Systems* (NeurIPS), 2014.

[14] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerry-Ryan, R. A. Saurous, Y. Agiomyrgiannakis and Y. Wu, "Natural TTS synthesis by conditioning wavenet on mel spectrogram predictions," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing* (ICASSP), 2018, pp. 4779-4783.

[15] K. Kumar, R. Kumar, T. de Boissiere, L. Gestin, W. Zhen Teoh, J. Sotelo, A. de Brebisson, Y. Bengio and A. Courville, "MelGAN: generative adversarial networks for conditional waveform synthesis," in *Proc. Advances in Neural Information Processing Systems* (NeurIPS), 2019.

[16] L. Wan, Q. Wang, A. Papir and I. L. Moreno, "Generalized end-to-end loss for speaker verification," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing* (ICASSP), 2018, pp. 4879-4883.

[17] A. Odena, C. Olah and J. Shlens, "Conditional image synthesis with auxiliary classifier GANs," in *Proc. International Conference on Machine Learning* (ICML), 2016, pp. 2642-2651.

[18] C. Veaux, J. Yamagishi and K. MacDonald, "CSTR VCTK corpus: English multi-speaker corpus for CSTR voice cloning toolkit," *University of Edinburgh. The Centre for Speech Technology Research* (CSTR), 2017.

[19] H. Zen, V. Dang, R. Clark, Y. Zhang, R. J. Weiss, Y. Jia, Z. Chen and Y. Wu, "LibriTTS: a corpus derived from Librispeech for text-to-speech," *arxiv.org/abs/1904.02882*, 2019.

[20] D. P. Kingma and J. Ba, "Adam: a method for stochastic optimization," in *Proc. International Conference on Learning Representations* (ICLR), 2015.

[21] M. Chen, X. Tan, B. Li, Y. Liu, T. Qin, S. Zhao and T.-Y. Liu, "Adaspeech: adaptive text to speech for custom voice," in *Proc. International Conference on Learning Representations* (ICLR), 2021.