INTERSPEECH 2023
20-24 August 2023, Dublin, Ireland

LightClone: Speaker-guided Parallel Subnet Selection for Few-shot Voice Cloning

Jie Wu, Jian Luan, Yujun Wang

Xiaomi AI Lab, Beijing, China
{wujie9, luanjian, wangyujun}@xiaomi.com

Abstract

Large-scale few-shot voice cloning service faces three main challenges: model storage for huge number of users, fast model training and real-time synthesis. They all involve model size directly. It is noted that few-shot voice cloning usually has much bigger model size than common TTS trained by one speaker corpus, since its source model needs more parameters to hold the characteristics of various speakers. It also indicates that a high quality TTS model for one voice could be much smaller. To reduce model size of voice cloning, speaker-guided parallel subnet selection (SG-PSS) is proposed in this paper. In adaptation phase, only one subnet is selected from parallel ones of source model for target speaker. By this method, adaptation training and inference can be much faster. Experiment results show that the proposed approach achieves 4x model compression ratio, 3x inference speedup and even slightly better performance in voice quality and speaker similarity in comparison with baseline.

Index Terms: few-shot speech synthesis, model compression, speaker-guided parallel subnet selection (SG-PSS)

1. Introduction

Recent years, with the rapid development of end-to-end text-to-speech (TTS) framework, e.g.,Tacotron [1, 2], Fastspeech [3, 4] and DurIAN [5], customized TTS (also called few-shot TTS, voice cloning) has made great progress, which is an important branch of traditional TTS to build a high-performance TTS system for a speaker with limited data, i.e., a few minutes or even only several samples. Nowadays, customized TTS has widespread application in intelligent customer service, news broadcast, audiobook narration, etc. Meanwhile, many speech platforms also provide customized TTS service to end users. As the demand of customized TTS increases rapidly, some challenges emerged, including less speech data collected from users, less storage cost for customized model and higher efficiency both in training and inference.

A popular customized TTS service usually needs to support millions of end users. For this reason, the training and inference efficiency of customized model are shed more light on reducing server cost. On the other hand, the footprint of each customized model should be as small as possible to save storage cost and loading time. Some prior works have been reported on designing lightweight and efficient TTS models. Luo [6] proposes a neural architecture search strategy to automatically design lightweight models with on par voice quality. DeviceTTS [7] leverages deep feedforward sequential memory network of small model size to accelerate inference speed. Depthwise separable convolution [8] and dynamic convolution [9] are leveraged to replace self-attention completely to reduce memory storage [10, 11]. However, the pre-trained source model for few-shot voice cloning requires enough number of parameters in neural networks to hold the characteristics of various speakers. Therefore, simply compacting the model size tends to degrade the quality of final customized voice seriously.

Meanwhile, many studies have been made on trading off the number of adaptation parameters and voice performance. Arik [12] and Chen [13] both conduct comparative experiments on finetuning only speaker embedding or the whole model. It shows that finetuning speaker embedding updates less parameters but has much poorer voice naturalness. BoffinNTTS [14] proves that freezing the weights of encoder leads to more robust synthesis. AdaDurIAN [15] evaluates the performance of finetuning different modules by calculating the word error rate and finds that fixing encoder and embedding modules (e.g. phoneme, tone, language and emotion) could achieve the least pronunciation errors. Meanwhile, AdaSpeech [16] introduces conditional layer normalization into the decoder of Fastspeech2. The scale and bias of the conditional layer normalization are predicted by a module from speaker embedding. In adaptation phase, AdaSpeech finetunes both the prediction module and the speaker embedding. It achieves high performance with few speaker specific parameters.

Generally, few-shot voice cloning usually build adaptation on a pre-trained multi-speaker model of a large number of trainable parameters. However, prior works also indicate that a high quality TTS model for one voice could be much smaller. Therefore, there might exists parameter redundancy in the adaptation phase. How to trim out redundant parameters before adaptation phase is the key factor to build a fast and light few-shot voice cloning. Shazeer [17] proposes a sparsely-gated mixture-of-experts (MOE) layer consisting of up to thousands of feed-forward sub-networks and a gating network determines a sparse combination of these experts with top-k values, which aims at dramatically increasing model capacity with only minor losses in computational efficiency. Inspired by it, we propose Speaker Guided Parallel Subnet Selection (SG-PSS) for few-shot voice cloning task. It splits a neural network layer into several parallel subnets and the speaker embedding is employed to control the output gate of each subnet. In source model training phase, the speaker characteristics are clustered into these parallel subnets. While in adaptation training and inference phase, only one subnet is selected among parallel ones based on speaker similarity. With the reduction of model size, it speeds up the adaptation training and real time synthesis effectively. The contributions of this paper include:

- Design parallel sub-nets architecture with gating network for speaker characteristic clustering.
- Introduce batch nuclear-norm maximization (BNM) loss to improve the discriminability and diversity of speaker charac-

10.21437/Interspeech.2023-1528

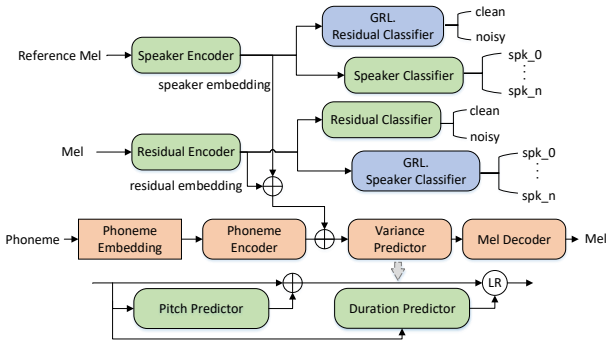

Figure 1: *The architecture of Fastspeech2-based TTS model including phoneme encoder, variance predictor, mel decoder, speaker encoder and residual encoder.*

teristic clustering.

- Propose several gating strategies based on speaker embedding for stacked multiple parallel sub-nets.
- Propose and evaluate subnet selection strategy for fast adaptation training and inference.

2. Baseline Architecture

As shown in Figure 1, Fastspeech2 [4] is adopted as the backbone of baseline network. To leverage multi-speaker training data, a speaker encoder is employed to extract speaker embedding from reference mel. In addition, a residual encoder is designed to capture the environment noise in target mel. Here, target mel represents the ground-truth mel which is used to calculate reconstruction mel loss. While reference mel is randomly selected from the same speaker of target mel. In this way, reference mel provides only speaker information and will not carry any content information of target mel. Classification loss and adversarial loss are both designed for two encoders to ensure the embedding correctness. They are:

$$L_{sc} = \sum_i L_{ce}(SC(s_i), s_{l_i}) \tag{1}$$

$$L_{adv_s} = \sum_i L_{ce}(RC(s_i), r_{l_i}) \tag{2}$$

$$L_{rc} = \sum_i L_{ce}(RC(r_i), r_{l_i}) \tag{3}$$

$$L_{adv_r} = \sum_i L_{ce}(SC(r_i), s_{l_i}) \tag{4}$$

where L_{ce} is cross entropy loss and SC, RC are speaker and residual classifiers. s_i and r_i represent the i-th speaker embedding and residual embedding, and s_l, r_l are its speaker identity and residual labels.

The total training loss is calculated as below:

$$\begin{aligned} L_G = &\lambda_{recon} * (L_1^{mel} + L_1^{pitch} + L_1^{dur}) \\ &+ \lambda_{spk} * (L_{sc} + L_{adv_s}) \\ &+ \lambda_{res} * (L_{rc} + L_{adv_r}) \end{aligned} \tag{5}$$

where L_1^{mel}, L_1^{pitch} and L_1^{dur} mean the L1 loss for predicted mel, pitch and duration respectively. λ_{recon}, λ_{spk} and λ_{res} are weights for losses of different modules.

In few-shot voice cloning stage, only the parameters of pitch predictor, duration predictor and 2-layer 1D convolutional network of each FFT block in decoder are updated. In addition, to enhance the training stability in adaptation stage, target

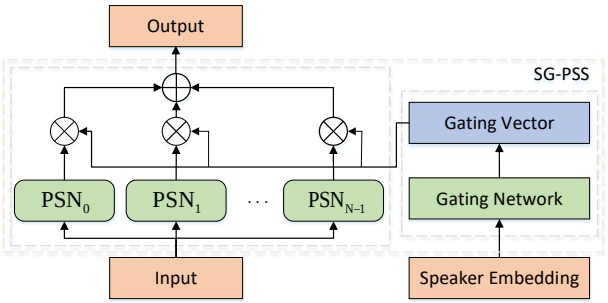

Figure 2: *The architecture of Speaker-guided Parallel Subnet Selection (SG-PSS).*

speaker embedding is represented by the nearest one in source model training corpus. The similarity of speaker embedding is measured by Euclidean distance.

3. Proposed Architecture

3.1. Speaker-guided Parallel Subnet Selection

As shown in Figure 2, the SG-PSS layer consists of N parallel subnets, PSN_0, PSN_1...PSN_{N-1}. Each of them has identical architecture (e.g., linear layer, convolutional layer), but with separate parameters. In pre-training phase, they are fed with the same input and their outputs are multiplied by respective gating vectors before added together. The gating vectors herein are one-hot like and determined by speaker embedding. It means similar speakers are likely to be concentrated on the same one subnet. Therefore, the training process can be regarded as an automatic speaker clustering as well and each subnet represents the characteristics of one speaker cluster. The gating network consists of two stacked dense layers and a softmax layer. The calculation of SG-PSS in pre-training is shown as below:

$$y = \sum_{i=0}^{N-1} PSN_i(x)(GN(spk\_emb))) \tag{6}$$

where x, y represent the input and output of SG-PSS module, GN stands for gating network, PSN_i denotes the i-th parallel sub-network and spk_emb is speaker embedding.

While in adaptation phase, the target speaker embedding is selected following the same selection strategy of baseline system. Then it is fed into the gating network to generate gates. Only the subnet corresponding to the greatest gate will be remained for fine-tuning and others are trimmed.Therefore, only $1/N$ parameters of SG-PSS layer are kept in adaptation training and final customized model, which results in reduction of computation cost and memory storage.

The structure of N parallel sub-networks in SG-PSS seems similar as the experts in MOE [17]. However, there are at least three differences between them. At first, their application purposes are different. MOE is designed to expand model parameters efficiently to improve model capability for very large data set (e.g., 100 billion word corpus in language modeling, machine translation and speech-to-animation task) [18]. On the right contrary, SG-PSS is proposed to compact model parameters for target speaker in voice cloning task. Secondly, for the input of gating network, MOE employs the same input as each experts while SG-PSS employs speaker embedding to achieve speaker clustering purpose. Thirdly, the gating network of MOE chooses top k out of N experts for each example, while SG-PSS accepts only one subnet for a target speaker in voice cloning.

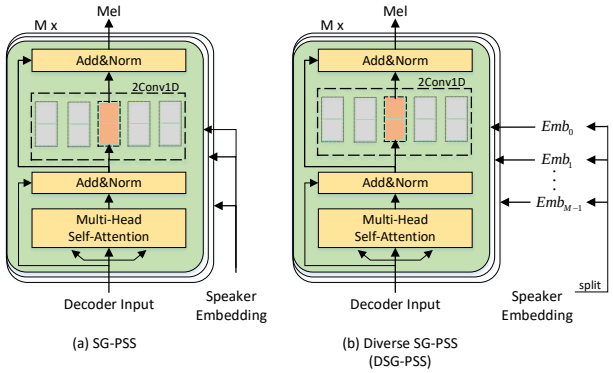

Figure 3: *A diagram of Speaker-guided Parallel Subnet Selection (SG-PSS) in stacked FFT blocks.*

3.2. SG-PSS in stacked FFT blocks

The decoder of Fastspeech2 stacks multiple feedforward transformer (FFT) blocks. Each FFT block consists of a multi-head self-attention layer and a 2-layer 1D convolutional network with ReLU activation. Since the decoder is regarded as more speaker-dependent, adaptation process usually fine-tunes the 2-layer 1D convolutional network of each FFT block in the decoder to approach the characteristics of target speaker. In this paper, SG-PSS is applied to the 2-layer 1D convolutional network in each FFT block. As shown on the left of Figure 3, the SG-PSS network consists of N parallel subnets. Each subnet is also a 2-layer 1D convolutional network but with smaller size.To guarantee the fairness, the number of parameters in SG-PSS is usually kept the same with the origin network. It means the filter size in each subnet of is $[D_h/N, D_o]$ when the original filter size is $[D_h, D_o]$.

For multiple stacked FFT blocks, one simple option is that all the FFT blocks share the same gating network, as shown in the left of Figure 3. Specifically, the D-dimensional speaker embedding is fed to the gating network to obtain N-dimensional gating vector. The same gating vector is employed by all FFT blocks. By this method, all possible combinations of selected subnets in different blocks can be N at most. Another option is to untie the gating network that each FTT block may have separate gating control. Ideally, the possible combinations of selected subnets in M FFT blocks increase to be N^M. However, in practice, the gating vectors are still quite similar for each block since their inputs are exactly the same speaker embedding. Its result is not presented in this paper due to its similar performance as the approach with the shared gating network. In order to enhance the diversity of gating vectors among stacked FFT blocks, another option called diverse SG-PSS (DSG-PSS) splits the speaker embedding into M segments as the input of each gating network respectively. As shown on the right of Figure 3, each segment Emb_i could be D/M dimensional. In this way, the gating vector for a FFT block is determined by only one segment of speaker embedding and the DSG-PSS in different blocks may focus on the different parts of speaker characteristics. It is very helpful to achieve the expected combination diversity.

3.3. Batch Nuclear-norm Maximization Loss

Since the SG-PSS can be regarded as speaker clustering in source model training, the gating vector is expected to meet two properties: discriminability and diversity. Herein, the discriminability means SG-PSS should focus on only one subnet for a sample. While the diversity means every subnet should be almost equally employed for all samples. For this purpose, batch nuclear-norm maximization (BNM) loss is introduced as follows:

$$L_{SG-PSS} = L_G + \lambda_{bnm} * L_{BNM} \tag{7}$$

The BNM loss is proposed by Cui [19] to boost the learning under typical label insufficient learning scenarios, such as semi-supervised learning, domain adaptation and open domain recognition. Herein, the gating vector represents the classification result which is SoftMax-ed output of gating network. For the gating vector GV of size N, its corresponding BNM loss is formulated as:

$$L_{BNM} = -\frac{1}{N}||GV||_* \tag{8}$$

$$||GV||_* = \sum_{i=1}^{D} \sigma_i \tag{9}$$

where $||GV||_*$ is nuclear-norm of gating vector GV, defined as the sum of singular value of GV. σ_i denotes the i-th largest singular value and D is the number of singular values.

4. Experiments

4.1. Dataset

The multi-speaker dataset including 89 Mandarin speakers (40 females and 49 males) is used for pre-training. Recordings are about 130 hours in total and collected in a professional recording studio. To mimic the environment noise of user recording, real environment noise is collected and randomly added to clean speech with SNR in the range of 25dB∼35dB. To evaluate the performance of few-shot voice cloning, 12 volunteers are recruited and 1 minute recording is collected on mobile in office environment for each of them. The phoneme duration of all recordings is obtained by Montreal Forced Aligner [20]. All recordings are processed to be 16kHz with 16bits per sample in mono channel. 80-dimensional mel-spectrogram and 1-dimensional $LF0$ are extracted at every 12.5ms. Moreover, there is no overlap between the pre-training and volunteers and text prompts for evaluation should be unseen both in pre-training and adaptation training stages.

4.2. Experimental setup

Both phoneme encoder and mel decoder consist of 6 FFT blocks following the basic configuration of Fastspeech [3]. Duration predictor and pitch predictor both have a 2-layer 1D convolutional network with 384 filters for each layer and a linear layer. In addition, the speaker encoder and residual encoder follow the Global Style Token (GST) architecture [21] with [48, 4] and [4, 2] for [tokens, attention heads] respectively. Speaker and residual classifier both contain two linear layers and a softmax layer. Besides, a gradient reversal layer (GRL) [22] is applied prior to corresponding classifier to do adversarial training. The multi-speaker model is trained with initial learning rate of 1×10^{-3} and batch size of 32. Adam optimizer is used with $\beta_1 = 0.9, \beta_2 = 0.98$. Dropout probability is 0.1 through the entire model. The training loss is L_G with $\lambda_{recon} = 5$, $\lambda_{spk} = 1$ and $\lambda_{res} = 1$.

In adaptation training phase, the parameters of duration predictor, pitch predictor and the 2-layer 1D convolutional networks in FFT blocks of mel decoder are fine-tuned 500 steps with batch size of 16. The efficient and high fidelity vocoder

HifiGAN [23] is adopted to generate waveform from predicted mel-spectrograms.

Four systems are implemented and evaluated as below:

Base-upper: Baseline system. Filter size of 2-layer 1D convolutional network in 6 FFT blocks is [2816, 384].

Base-lower: Similar as baseline, but the filter size of 2-layer 1D convolutional network is reduced to [704, 384], which is the same as one subnet in SG-PSS.

SG-PSS: Proposed approach. 6 stacked FFT blocks share one gating network shown in left of Figure 3. The SG-PSS has 4 parallel 2-layer 1D convolutional network with [704, 384] as filter size. The model size of SG-PSS equals to the Base-upper in pre-training while equals to the Base-lower in adaptation training. The training loss is L_{SG-PSS} with $\lambda_{bnm} = 0.5$.

DSG-PSS: Based on SG-PSS, 6 stacked FFT blocks have separate gating networks shown in right of Figure 3. The 384-dimensional speaker embedding is split into 6 segments of 64-dimensional vector as the input of 6 gating networks.

Table 1: *The comparisons of object metrics for four systems.*

	#Params[1] (multi-speaker)	#Params[1] (adaptation/inference)	RTF[2]	MCD[3]
Base-lower	11.873M	11.873M	0.126	5.053
Base-upper	41.083M	41.083M	0.382	4.957
SG-PSS	41.083M	11.873M	0.126	4.962
DSG-PSS	41.083M	11.873M	0.126	4.921

[1] #Params means the number of parameters for finetuning modules.
[2] RTF is measured over the entire model using a single thread and a single core on an Intel Xeon Gold 6240 CPU @ 2.60GHz.
[3] MCD represents the distortion between predicted and groudtruth mel-spectrograms. The smaller, the better.

4.3. Evaluation

4.3.1. Objective Evaluation

To measure the model storage, inference speed and synthesis quality of different systems, the number of fine-tuned parameters, real time factor (RTF) and mel-cepstrum distortion (MCD) are shown in Table 1. In terms of the number of fine-tuned parameters, the multi-speaker source model of SG-PSS and DSG-PSS have the same model size as Base-upper but larger than Base-lower. While in final adapted model, SG-PSS and DSG-PSS achieve almost 4x compression ratio and has the same model size with Base-lower. Correspondingly, in terms of RTF on CPU, SG-PSS, DSG-PSS and Base-lower is about 3x speedup in comparison with Base-upper. Further, in terms of MCD, DSG-PSS achieves the best synthesis quality while Base-Lower is the worst. SG-PSS and Base-upper are on par.

The predicted mel-spectrograms of a randomly selected sample by four systems are visualized in Figure 4. As shown in the green rectangles, the mel-sepctrogram of Base-lower is mostly fuzzy, while Base-upper is much clearer. This phenomenon can be attributed to the roust source model with larger parameters. The performance of SG-PSS is comparable to Base-upper. While the DSG-PSS generates the clearest harmonics. It proves that enhancing the diversity of subnet combination among stacked FFT blocks can benefit the model capability.

4.3.2. Subjective Evaluation

To evaluate the performance of voice cloning, subjective listening tests (mean opinion score, MOS) are conducted on voice quality and speaker similarity. We generate 22 samples per system and invite 15 native listeners to evaluate all systems.

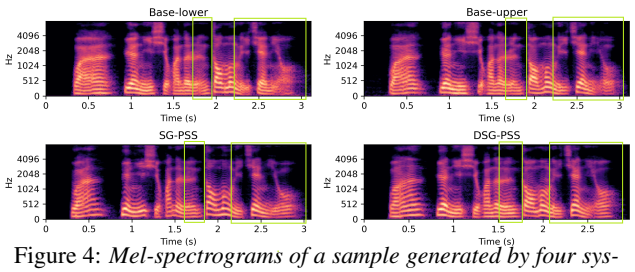

Figure 4: *Mel-spectrograms of a sample generated by four systems.*

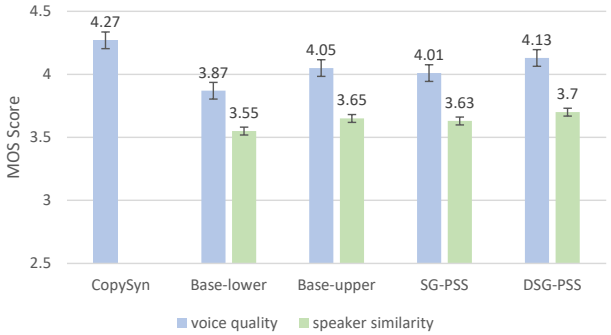

Figure 5: *Mean Opinion Score (MOS) test results of voice quality and speaker similarity for customized TTS voice performance with* 95% *confidence intervals.*

As shown in Figure 5, DSG-PSS achieves the highest MOS scores both in voice quality and speaker similarity. In addition, SG-PSS and Base-upper are almost on par and better than Base-lower obviously. Moreover, the feedback of listeners indicates that DSG-PSS can produce more natural rhythm, more stable tone and clearer articulation of vowels in various text context. It confirms the effectiveness of SG-PSS and separate gating strategy for different blocks. The results of subjective evaluation are consistent with MCD in Table 1 and the mel-spectrograms illustration shown in Figure 4.

5. Conclusions

In this paper, we propose speaker-guided parallel subnet selection (SG-PSS) to build a fast and light model for few-shot voice cloning task. Parallel subnets gated by speaker embedding act as speaker clustering in source model training. Batch nuclear-norm maximization (BNM) loss is introduced to improve the discriminability and diversity of the speaker clustering. In combination with speaker embedding segmentation and separate gating networks, the stacked multiple parallel subnets in decoder can effectively model large number of speaker characteristics with limited parameters. Furthermore, for a coming target speaker, it is easy to select the subnet he/she belongs to by speaker embedding and trim others. Therefore the final model size is compressed and inference speed is speedup. Experiment results show the best proposed approach (DSG-PSS) can achieve 4x model compression ratio and 3x speedup in adaptation training and inference speed when it even performs slightly better in speaker similarity and voice quality than the baseline. This paper applies the proposed SG-PSS and DSG-PSS into stacked FFT blocks for voice cloning task. However it is not difficult to image that it can be applied into other layer types or network architectures, and also benefit other tasks. Some samples for subjective evaluation are available via this link[1].

[1] https://lightclone2023.github.io/INTERSPEECH2023-demo/

6. References

[1] Y. Wang, R. Skerry-Ryan, D. Stanton, Y. Wu, R. J. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio *et al.*, "Tacotron: Towards end-to-end speech synthesis," *Proc. Interspeech 2017*, pp. 4006–4010, 2017.

[2] R. Skerry-Ryan, E. Battenberg, Y. Xiao, Y. Wang, D. Stanton, J. Shor, R. Weiss, R. Clark, and R. A. Saurous, "Towards end-to-end prosody transfer for expressive speech synthesis with tacotron," in *international conference on machine learning*. PMLR, 2018, pp. 4693–4702.

[3] Y. Ren, Y. Ruan, X. Tan, T. Qin, S. Zhao, Z. Zhao, and T.-Y. Liu, "Fastspeech: Fast, robust and controllable text to speech," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[4] 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 *International Conference on Learning Representations*, 2020.

[5] C. Yu, H. Lu, N. Hu, M. Yu, C. Weng, K. Xu, P. Liu, D. Tuo, S. Kang, G. Lei *et al.*, "Durian: Duration informed attention network for multimodal synthesis," *Proc. Interspeech 2020*, 2020.

[6] R. Luo, X. Tan, R. Wang, T. Qin, J. Li, S. Zhao, E. Chen, and T.-Y. Liu, "Lightspeech: Lightweight and fast text to speech with neural architecture search," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 5699–5703.

[7] Z. Huang, H. Li, and M. Lei, "Devicetts: A small-footprint, fast, stable network for on-device text-to-speech," *arXiv preprint arXiv:2010.15311*, 2020.

[8] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1251–1258.

[9] F. Wu, A. Fan, A. Baevski, Y. Dauphin, and M. Auli, "Pay less attention with lightweight and dynamic convolutions," in *International Conference on Learning Representations*, 2018.

[10] S. Li, B. Ouyang, L. Li, and Q. Hong, "Light-tts: Lightweight multi-speaker multi-lingual text-to-speech," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 8383–8387.

[11] L. Kaiser, A. N. Gomez, and F. Chollet, "Depthwise separable convolutions for neural machine translation," in *International Conference on Learning Representations*, 2018.

[12] S. Ö. Arık, J. Chen, K. Peng, W. Ping, and Y. Zhou, "Neural voice cloning with a few samples," in *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, 2018, pp. 10 040–10 050.

[13] Y. Chen, Y. Assael, B. Shillingford, D. Budden, S. Reed, H. Zen, Q. Wang, L. C. Cobo, A. Trask, B. Laurie *et al.*, "Sample efficient adaptive text-to-speech," in *International Conference on Learning Representations*, 2018.

[14] H. B. Moss, V. Aggarwal, N. Prateek, J. González, and R. Barra-Chicote, "Boffin tts: Few-shot speaker adaptation by bayesian optimization," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 7639–7643.

[15] Z. Zhang, Q. Tian, H. Lu, L.-H. Chen, and S. Liu, "Adadurian: Few-shot adaptation for neural text-to-speech with durian," *arXiv preprint arXiv:2005.05642*, 2020.

[16] M. Chen, X. Tan, B. Li, Y. Liu, T. Qin, T.-Y. Liu *et al.*, "Adaspeech: Adaptive text to speech for custom voice," in *International Conference on Learning Representations*, 2020.

[17] N. Shazeer, A. Mirhoseini, K. Maziarz, A. Davis, Q. Le, G. Hinton, and J. Dean, "Outrageously large neural networks: The sparsely-gated mixture-of-experts layer," *ICLR*, 2017.

[18] L. Chen, Z. Wu, J. Ling, R. Li, X. Tan, and S. Zhao, "Transformer-s2a: Robust and efficient speech-to-animation," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 7247–7251.

[19] S. Cui, S. Wang, J. Zhuo, L. Li, Q. Huang, and Q. Tian, "Towards discriminability and diversity: Batch nuclear-norm maximization under label insufficient situations," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 3941–3950.

[20] M. McAuliffe, M. Socolof, S. Mihuc, M. Wagner, and M. Sonderegger, "Montreal forced aligner: Trainable text-speech alignment using kaldi." in *Interspeech*, vol. 2017, 2017, pp. 498–502.

[21] Y. Wang, D. Stanton, Y. Zhang, R.-S. Ryan, E. Battenberg, J. Shor, Y. Xiao, Y. Jia, F. Ren, and R. A. Saurous, "Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis," in *International Conference on Machine Learning*. PMLR, 2018, pp. 5180–5189.

[22] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, "Domain-adversarial training of neural networks," *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 2096–2030, 2016.

[23] J. Kong, J. Kim, and J. Bae, "Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis," *Advances in Neural Information Processing Systems*, vol. 33, pp. 17 022–17 033, 2020.