INTERSPEECH 2023
20-24 August 2023, Dublin, Ireland

Pruning Self-Attention for Zero-Shot Multi-Speaker Text-to-Speech

Hyungchan Yoon[1], *Changhwan Kim*[1], *Eunwoo Song*[2], *Hyun-Wook Yoon*[2], *Hong-Goo Kang*[1]

[1]Department of Electrical and Electronic Engineering, Yonsei University, Seoul, Korea,
[2]NAVER Cloud, South Korea

[hcy71, chkim]@dsp.yonsei.ac.kr, [eunwoo.song, hyunwook.yoon]@navercorp.com,
hgkang@yonsei.ac.kr

Abstract

For personalized speech generation, a neural text-to-speech (TTS) model must be successfully implemented with limited data from a target speaker. To this end, the baseline TTS model needs to be amply generalized to out-of-domain data (i.e., target speaker's speech). However, approaches to address this out-of-domain generalization problem in TTS have yet to be thoroughly studied. In this work, we propose an effective pruning method for a transformer known as *sparse attention*, to improve the TTS model's generalization abilities. In particular, we prune off redundant connections from self-attention layers whose attention weights are below the threshold. To flexibly determine the pruning strength for searching optimal degree of generalization, we also propose a new differentiable pruning method that allows the model to automatically learn the thresholds. Evaluations on zero-shot multi-speaker TTS verify the effectiveness of our method in terms of voice quality and speaker similarity.
Index Terms Text-to-speech, zero-shot, generalization, sparse attention

1. Introduction

With the advancement of deep learning technologies, recent studies in text-to-speech (TTS) have shown a rapid progress. In terms of generation quality, single- and multi-speaker TTS models can synthesize human-like voices with sufficient training data from the target speaker(s) [1–5]. Further, several few- or zero-shot multi-speaker TTS models have recently been developed to synthesize out-of-domain (OOD) speech with limited data from the target speaker [6–11]. These models are trained using a large multi-speaker dataset to learn a general TTS mapping relationship conditioned on speaker representations. Then, they are either additionally fine-tuned with a few samples of the target speaker (few-shot) or used directly (zero-shot) for synthesis.

Especially, zero-shot multi-speaker TTS models [8–11] are widely being studied due to their unique advantage of not requiring any training data from the target speaker. A common approach of these models is to extract the speaker representations from reference speech using a reference encoder [7, 12, 13]. These representations contain various prosodic characteristics such as pronunciation style, speed [14, 15] of the reference speech, as well as speaker identity. As such, the speaker representation is learned to play a crucial role as a latent vector that determines the prosodic characteristics of the synthesized speech during training. During inference, the speaker representation is extracted from the voice of the unseen speaker, enabling the generation of the desired voice.

However, zero-shot multi-speaker TTS models face the problem of domain mismatch between training and inference, unlike conventional TTS models that aim to synthesize only in-domain speech (i.e., speech from seen speakers). Specifically, the latter must be generalized only to the unseen text, whereas the former must generalized not only to the unseen text but also to the reference speech of unseen speakers. Therefore, the challenge of improving synthesis performance in zero-shot multi-speaker TTS lies in generalizing the TTS models to OOD data, which refers to speech from unseen speakers.

One additional challenge faced by zero-shot multi-speaker TTS models is that they require varying levels of generalization ability depending on the dataset they are trained on. When there is a high degree of domain mismatch between the training and test data, such as differences in recording environments, the models require more generalization to prevent overfitting. Conversely, when there is little domain mismatch, over-generalization can lead to degraded performance. Therefore, finding the optimal strength of generalization is crucial for improving the synthesis performance of these models. However, current zero-shot multi-speaker TTS models lack a systematic approach to this problem and have difficulty controlling the generalization strength once developed. While adjusting the number of parameters is a classical approach to controlling generalization [16], it can be a manual and time-consuming process.

To this end, we propose a new controllable generalization method for zero-shot multi-speaker TTS models. In particular, we focus on the transformer [17], which is the foundation for many TTS models. Our method draws on previous studies in various research fields (such as image generation and speech recognition) demonstrating the effectiveness of optimizing the self-attention module in a generalization objective [18–22]. In particular, they enhanced generalization abilities by adding sparsity to the self-attention connections. For instance, Child et al. [18] factorized the self-attention matrix into sparse subsets, and Kim et al. [21] proposed removing the low-weight connections during inference.

In this study, we design a sparse attention method for zero-shot multi-speaker TTS to successfully solve its OOD generalization problem. The method is implemented by pruning off the connections from self-attention layer; we also propose a *differentiable pruning* technique that can easily control the degree of generalization. Our contributions are outlined below:

- **New Application.** We apply the sparse attention mechanism to the TTS model, which eliminates redundant connections from the self-attention layer. Because the TTS model is trained under a condition that only uses high-weight residual connections, the sparse attention mechanism significantly improves its generalization ability. In particular, adding sparsity to the self-attention module reduces the number of parameters engaged in the overall TTS training by preventing backpropagation of gradients through low-weight connections, which alleviates overfitting.

10.21437/Interspeech.2023-1301

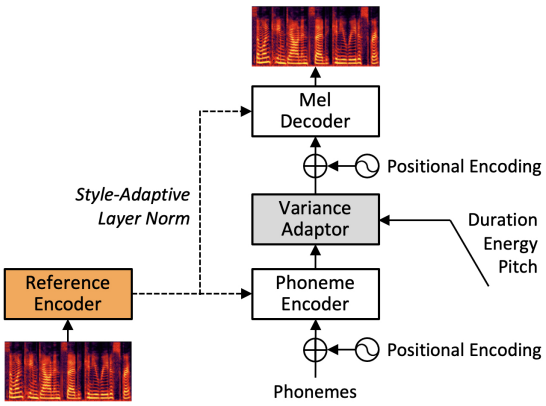


Figure 1: Overview of StyleSpeech. The speaker representation is extracted from the reference encoder and provided to the encoder and decoder via Style-Adaptive Layer Normalization technique.

- **Novel Pruning Technique.** We explore optimal pruning techniques for the sparse attention. We first introduce a vanilla pruning approach that eliminates the connections whose attention weights are below a predetermined threshold. To flexibly adjust the pruning strength in case of various degrees of domain mismatch, we further propose a differentiable pruning method that adopts learnable thresholds.
- **Performance.** Experiments on zero-shot TTS show that our proposed method notably improves the performance of OOD speech synthesis[1].

2. Related Works

Owing to the increasing demand for customized voice synthesis, the OOD generalization problem has recently been studied in zero-shot multi-speaker TTS works. StyleSpeech [7] used meta-learning to make a TTS model effectively adapt to OOD voice and conditioned speaker representations to the model using few variables to minimize the domain mismatch. For the same purpose, nnSpeech [8] introduced a speaker-guided conditional varational autoencoder to define speaker representations as Gaussian latent variables rather than high-dimensional embeddings. Furthermore, GenerSpeech [11] leveraged wav2vec2.0 [23], a contrastive model learned with numerous speech data, to obtain more robust speaker representations. Unlike the abovementioned approaches, we use the self-attention pruning method to directly generalize the basic architecture (i.e., transformer) of the TTS model, implying that it is applicable to other models with minimal modifications.

3. Proposed Method

We selected StyleSpeech [7] as a baseline because it is a representative zero-shot multi-speaker TTS model built on a non-autoregressive transformer. As depicted in Fig 1, its architecture comprises a transformer-based phoneme encoder and mel-spectrogram decoder, a variance adaptor, and a reference encoder. The variance adaptor, located between the encoder and decoder, predicts the pitch, energy, and duration from phoneme-level embeddings; it then expands these embeddings to frame-level using the predicted duration values. The reference encoder extracts a speaker representation from the input reference speech and conditions it to the encoder and decoder via Style-Adaptive Layer Normalization [7] technique. More details, including loss terms and model configurations, are presented in [2, 7].

[1]Audio samples are available at: `https://hcy71o.github.io/SparseTTS-demo/`

3.1. Sparse Attention

We implement sparse attention by pruning redundant connections, and we only apply it to the decoder for the following two reasons: 1) The sequence length (N) of the decoder (frame-level) is much longer than that of the encoder (phoneme-level), indicating that the decoder has a significantly larger number of self-attention connections ($N \times N$) than the encoder; as a result, the decoder self-attention module requires more sparsity to be generalized. 2) According to our investigation, applying sparse attention to the encoder rather degrades the model performance because it reduces the modeling capacity of the original self-attention module. We define sparse masks and apply them to all the attention heads of the decoder self-attention modules. Depending on the mask generation methods, we propose two types of pruning techniques: **vanilla** and **differentiable**.

3.1.1. Vanilla Pruning

Given queries Q and keys K obtained by two linear transformations W_q and W_k, respectively, to the input sequence $\mathbf{x}$,

$$Q = W_q\mathbf{x},\ K = W_k\mathbf{x}, \tag{1}$$

we first denote the attention probability of the h-th head of the multi-head self-attention layer [17] as $\mathcal{A}_h$:

$$\mathcal{A}_h(i,j) = softmax\left(\frac{Q_h {K_h}^T}{\sqrt{d}}\right)_{(i,j)}, \tag{2}$$

where Q_h and K_h are the queries and keys of the h-th head, respectively, and d is their dimension. $\mathcal{A}_h(i,j)$ indicates the weight score of the i-th query corresponding to the j-th key. We then define a sparse mask matrix SM^h of h-th head as follows:

$$SM^h_{(i,j)} = \begin{cases} 1 & \text{if } \mathcal{A}_h(i,j) \geq \mu_i \\ 0 & \text{if } \mathcal{A}_h(i,j) < \mu_i \end{cases}, \tag{3}$$

$$\mu_i = \frac{1}{N}\sum_{j=1}^{N}\mathcal{A}_h(i,j), \tag{4}$$

where N is the length of the input sequence $\mathbf{x}$. Applied to $\mathcal{A}_h$, the SM^h mask prunes its weak connections, whose weights are below the average attention weights μ_i along the key axis.

During the experiment, we observed that using a common sparse mask combined along the head-axis outperforms applying SM^h to each head individually. In detail, we consider each head's activated positions for all the other heads; we define an adjusted sparse mask $SM_{OR} := \bigcup_{h=1}^{H} SM^h$ where H is the number of heads, and identically apply it to all attention heads. The SM_{OR} mask is used during both training and inference.

3.1.2. Differentiable Pruning

In the vanilla pruning (VP) method, the threshold of SM^h is passively determined as the mean value of attention weights μ_i. However, the optimal threshold values vary depending on the number of layers, type of generation tasks, and degree of domain mismatch; thus, flexibly setting the threshold is preferable. To this end, we propose a novel differentiable pruning

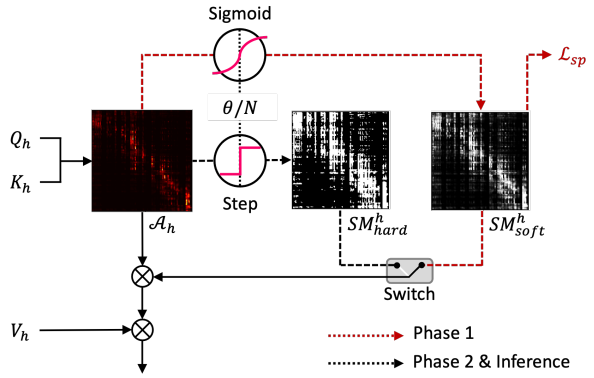


Figure 2: Overview of the differentiable pruning. The soft sparse mask is used during phase 1 of training, and the hard sparse mask is used during phase 2 of training and during inference. The pruned attention head is multiplied with V_h (values of h-th head) for succeeding process of self-attention operation.

(DP) method with learnable thresholds, inspired by [22] which shares the same motivation in natural language processing task.

Fig. 2 illustrates the overview of DP. In contrast to VP, which uses the predefined threshold, we first define a hard sparse mask of h-th head SM^h_{hard} that inherits the learnable threshold θ/N:

$$SM^h_{hard(i,j)} = \begin{cases} 1 & \text{if } \mathcal{A}_h(i,j) \geq \theta/N \\ 0 & \text{if } \mathcal{A}_h(i,j) < \theta/N \end{cases}, \tag{5}$$

where θ is a trainable threshold parameter, and N is the length of the sequence used to adjust the threshold value based on variations in input length. However, because the process of obtaining the binary mask SM^h_{hard} is not differentiable, we cannot directly update θ by gradient descent. To solve this problem, we additionally adopt a differentiable soft sparse mask SM^h_{soft} defined by a sigmoid function as follows:

$$SM^h_{soft} = \sigma\left(\frac{\mathcal{A}_h - \theta/N}{T}\right), \tag{6}$$

where T is the temperature set to 0.01 to approximate SM^h_{soft} to SM^h_{hard}. The value of SM^h_{soft} is close to 1 where the attention weight is higher than the threshold θ/N and is close to 0 in the opposite case.

We then propose a two-phase training method, summarized in Algorithm 1. In phase 1, the entire model is trained with original TTS loss terms $\mathcal{L}_{tts}$ (i.e., loss terms of StyleSpeech [7]) using the soft sparse masks SM_{soft} to update model parameters including thresholds θ. We also add sparsity loss $\mathcal{L}_{sp}$ as a regularization term to ensure pruning behavior, as shown below:

$$\mathcal{L}_{sp} = \frac{1}{LH}\sum_{l=1}^{L}\sum_{h=1}^{H}\left(\overline{SM}^h_{soft} - R\right)^2, \tag{7}$$

where the sparsity ratio R is a hyperparameter that indirectly determines the pruning strength; L denotes the number of transformer layers, and H represents the number of heads in each layer. Sparsity loss $\mathcal{L}_{sp}$ is defined as the average $L2$-distance between the soft sparse mask's mean value $\overline{SM}^h_{soft}$ and R across all attention heads and decoder layers. This loss term forces the model to be generalized to OOD data; without it, thresholds θ do not converge to meaningful values during training. This is because if only $\mathcal{L}_{tts}$ is used, the model obtains the lowest training loss value when *no connections are pruned* (i.e., θ is stuck in 0) for *in-domain* data. The value of R is set between 0 and 1, and a lower R value prunes more connections. In summary, two loss terms are used when updating θ: original TTS loss terms $\mathcal{L}_{tts}$ and the regularization term $\mathcal{L}_{sp}$. As mentioned previously, $\mathcal{L}_{tts}$ pulls θ down towards 0, while $\mathcal{L}_{sp}$ aims to prevent this for generalization to OOD data. The threshold is consequently balanced by two opposing losses, pruning off self-attention connections only to the extent that it does not significantly harm the original objective of minimizing $\mathcal{L}_{tts}$. Thus, the degree of generalization can be controlled by varying the sparsity ratio R, while minimizing overgeneralization.

Algorithm 1 Differentiable pruning - Training procedure

Phase 1: Apply the soft sparse masks SM_{soft}, and update the model parameters including the thresholds θ with the original TTS loss terms $\mathcal{L}_{tts}$ and additional regularization term $\mathcal{L}_{sp}$

Phase 2: Apply the hard sparse masks SM_{hard} determined by the learned thresholds θ, and update the residual model parameters with training loss terms except $\mathcal{L}_{sp}$

In phase 2, model parameters except θ are updated using the hard sparse masks SM_{hard}, whose thresholds θ are learned in phase 1. Here, $\mathcal{L}_{sp}$ is not used and the model is trained under the *hard* pruning condition (low-weight connections are completely masked) with fixed pruning strength. This final model is used during inference.

4. Experiments

4.1. Experimental Setup

Dataset and Preprocessing. We used two subset datasets from LibriTTS [24] (*train-clean-100* and *train-clean-360*) to train our model, which contain 245 hours of speech from 1151 speakers. For inference, VCTK [25] (108 unseen speakers) dataset is used for zero-shot TTS. For the method of preprocessing text and speech, we followed StyleSpeech [7].

Model Details. We experimentally evaluated the performance of VP and DP considering StyleSpeech [7] as a baseline. Consistent with [7], the encoder and decoder comprise 4 FFT blocks [2] with 2 self-attention heads each. For DP, 4 threshold parameters θ were declared in each decoder FFT block and were identically initialized to 0. The training configurations for all implemented models were set to be the same as in [7], except that the models were trained for 300k steps. In the case of DP, we advanced to phase 2 after training for 40k steps in phase 1. For evaluation, we used the HiFi-GAN V1 vocoder [26] to convert mel-spectrograms to audios. In addition, two references were used for comparison: 1) ground truth audios and 2) audios generated by HiFi-GAN V1 (Voc.) conditioned on ground truth mel-spectrograms.

Evaluation Metrics. Regarding subjective metrics, mean opinion score (MOS) evaluates the naturalness of speech, and similarity MOS (SMOS) evaluates speaker similarity. Both metrics were scored on a scale of 1-5 by 16 raters, and we present them with 95% confidence intervals (CI). We used character error rate (CER) and speaker embedding cosine similarity (SECS) to evaluate intelligibility and speaker similarity as objective metrics. For CER, we transcribed the synthesized speech using the pre-trained speech recognition model provided by the SpeechBrain toolkit [27]. SECS is defined as the cosine similarity between speaker embeddings derived from the pre-trained speaker ver-

Table 1: Comparisons of MOS, SMOS with 95% CI, CER, and SECS results of zero-shot TTS; we used the StyleSpeech framework as the baseline system. Note that VP and DP denote the vanilla and differentiable pruning techniques, respectively. For DP, we conducted 3 experiments by varying the sparsity ratio R. The best performances are in boldface.

Model	MOS(↑)	SMOS(↑)	CER(↓)	SECS(↑)
Ground Truth	4.76±0.07	-	-	-
GT mel + Voc.	4.67±0.08	-	-	-
Baseline	3.43±0.12	2.99±0.16	4.56	0.268
VP	3.46±0.12	3.10±0.15	5.17	0.275
DP($R = 0.50$)	3.53±0.12	3.18±0.15	3.96	**0.279**
DP($R = 0.45$)	**3.76±0.11**	**3.23±0.15**	3.96	0.278
DP($R = 0.40$)	3.75±0.12	3.20±0.16	**3.73**	0.276

Table 2: MOS, SMOS with 95% CI, CER, and SECS results of ablation studies. The best performances are in boldface.

Model	MOS(↑)	SMOS(↑)	CER(↓)	SECS(↑)
DP($R = 0.45$)	**3.76±0.11**	**3.23±0.15**	**3.96**	**0.278**
w/o SM_{hard}	3.65±0.11	3.02±0.16	4.21	0.274
w/o $\mathcal{L}_{sp}$	3.46±0.12	2.87±0.15	5.77	0.263

ification model [28] from [27]. Thus, MOS and CER assess speech quality, whereas SMOS and SECS assess similarity to the target speaker.

4.2. Evaluation on Zero-Shot TTS

For zero-shot TTS, we used arbitrary text input and randomly sampled one reference speech from each VCTK speaker for the reference encoder's input. 15 synthesized samples were used for MOS and SMOS, and 100 samples were used for CER and SECS.

From Table 1, we make the following observations: 1) The model with VP outperforms the baseline in all metrics except CER, demonstrating the generalization ability of the pruning method. 2) All models with DP remarkably surpass the baseline and the model with VP, particularly in terms of voice quality. 3) The results among models with DP show the trade-off relationship between pruning strength and performance. In the first viewpoint, the model is successfully generalized by pruning more connections ($R : 0.50 \rightarrow 0.45$), resulting in a sharp increase in naturalness (+0.23 MOS). In contrast, excessive pruning ($R : 0.45 \rightarrow 0.40$) rather reduces the model's original modeling capacity (i.e., overgeneralization); it causes a slight degradation in overall performance in our experiment. Intuitively, pruning all connections is the same as removing the entire self-attention module.

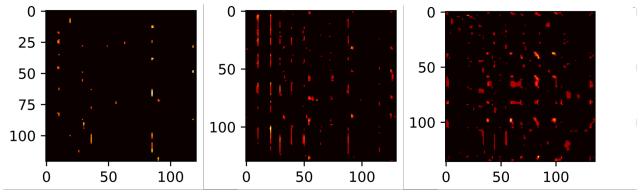
In summary, we conclude that DP significantly improves zero-shot TTS performance. Owing to its ability to adjust pruning strength, the model is also scalable to different degrees of domain mismatch (e.g., small R in large domain mismatch).

4.3. Ablation Study

Table 2 shows the results of the ablation studies related to the two DP design techniques. We chose DP with $R = 0.45$ as the baseline because it performs best in terms of naturalness and similarity. In the first experiment, we skipped the training phase 2 that uses the hard masks SM_{hard}; we only used the soft masks SM_{soft} for training and inference. Results show that the two-phase training method is effective. Concretely, in phase 2, *hard* pruning with updated thresholds improves the model's generalization performance by completely excluding low-weight connections during the text-to-mel conversion process. In the second experiment, we removed the regularization term $\mathcal{L}_{sp}$, originally used in the training phase 1. Without $\mathcal{L}_{sp}$, the model shows poor performance because pruning does not occur at all. We also discovered that the thresholds θ were not updated from their initial value of 0, as noted in section 3.1.2.

Table 3: Final DP thresholds θ updated in training phase 1.

Model	Threshold θ			
	Layer #1	Layer #2	Layer #3	Layer #4
DP($R = 0.50$)	0.76	2.34	2.36	2.36
DP($R = 0.45$)	1.70	3.53	4.18	4.18
DP($R = 0.40$)	2.89	3.05	5.11	5.11



(a) $R = 0.40$ (b) $R = 0.45$ (c) $R = 0.50$

Figure 3: *Pruned attention heads for the utterance "How many attention connections are pruned?". Samples are first heads of the fourth decoder layers and are generated from DP with (a)* $R = 0.40$*, (b)* $R = 0.45$*, and (c)* $R = 0.50$*.*

4.4. Analysis of Differentiable Pruning

To further analyze DP, we present the updated final thresholds θ of models with DP in Table 3. As expected, a smaller R value generally leads to higher threshold values, indicating that more connections are pruned. Fig. 3 represents the pruned attention heads of theses models using a specific text utterance and random reference speech. The previously mentioned relationship between R and pruning strength is also confirmed in the figure. Remarkably, the pruned TTS models use only a few self-attention connections for high synthesis quality, implying that DP prevents the decoder from overfitting to in-domain data and improves the generalization performance. More materials of visualizations are in our demo page.

5. Conclusion

In this work, we proposed a self-attention pruning method for improving the generalization abilities of zero-shot multi-speaker TTS models. Furthermore, we investigated the optimal pruning techniques and emphasized the importance of differentiable pruning (DP), that can control the pruning strength augmented with the proposed two-phase training method. We then used it to generalize the mel-spectrogram decoder; evaluation on zero-shot multi-speaker TTS confirmed its superiority in terms of voice quality and speaker similarity. Future works include the application of DP for more severe domain mismatch cases.

6. Acknowledgement

This work was supported by Voice&Avatar, NAVER Cloud, Seongnam, Korea.

7. 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 *Proc. ICASSP*, 2018, pp. 4779–4783.

[2] 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*, 2020.

[3] M. Chen, X. Tan, Y. Ren, J. Xu, H. Sun, S. Zhao, and T. Qin, "Multispeech: Multi-speaker text-to-speech with transformer," in *Proc. INTERSPEECH*, 2020.

[4] J. Kim, S. Kim, J. Kong, and S. Yoon, "Glow-TTS: A generative flow for text-to-speech via monotonic alignment search," in *Proc. NIPS*, vol. 33, 2020, pp. 8067–8077.

[5] 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.

[6] M. Chen, X. Tan, B. Li, Y. Liu, T. Qin, T.-Y. Liu *et al.*, "Adaspeech: Adaptive text-to-speech for custom voice," in *Proc. ICLR*, 2020.

[7] D. Min, D. B. Lee, E. Yang, and S. J. Hwang, "Meta-stylespeech: Multi-speaker adaptive text-to-speech generation," in *Proc. ICML*, 2021, pp. 7748–7759.

[8] B. Zhao, X. Zhang, J. Wang, N. Cheng, and J. Xiao, "nnspeech: Speaker-guided conditional variational autoencoder for zero-shot multi-speaker text-to-speech," in *Proc. ICASSP*, 2022, pp. 4293–4297.

[9] Y. Wu, X. Tan, B. Li, L. He, S. Zhao, R. Song, T. Qin, and T.-Y. Liu, "Adaspeech 4: Adaptive text-to-speech in zero-shot scenarios," in *Proc. INTERSPEECH*, 2022.

[10] M. Kim, M. Jeong, B. J. Choi, S. Ahn, J. Y. Lee, and N. S. Kim, "Transfer learning framework for low-resource text-to-speech using a large-scale unlabeled speech corpus," in *Proc. INTERSPEECH*, 2022.

[11] R. Huang, Y. Ren, J. Liu, C. Cui, and Z. Zhao, "Generspeech: Towards style transfer for generalizable out-of-domain text-to-speech synthesis," in *Proc. NIPS*, 2022.

[12] Y. Jia, Y. Zhang, R. Weiss, Q. Wang, J. Shen, F. Ren, P. Nguyen, R. Pang, I. Lopez Moreno, Y. Wu *et al.*, "Transfer learning from speaker verification to multispeaker text-to-speech synthesis," in *Proc. NIPS*, vol. 31, 2018.

[13] C.-M. Chien, J.-H. Lin, C.-y. Huang, P.-c. Hsu, and H.-y. Lee, "Investigating on incorporating pretrained and learnable speaker representations for multi-speaker multi-style text-to-speech," in *Proc. ICASSP*, 2021.

[14] 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 *Proc. ICML*, 2018, pp. 5180–5189.

[15] 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 *Proc. ICML*, 2018, pp. 4693–4702.

[16] C. L. Giles and C. W. Omlin, "Pruning recurrent neural networks for improved generalization performance," *IEEE transactions on neural networks*, vol. 5, no. 5, pp. 848–851, 1994.

[17] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. NIPS*, vol. 30, 2017.

[18] R. Child, S. Gray, A. Radford, and I. Sutskever, "Generating long sequences with sparse transformers," *ArXiv*, vol. abs/1904.10509, 2019.

[19] Y. Tay, D. Bahri, L. Yang, D. Metzler, and D.-C. Juan, "Sparse sinkhorn attention," in *ICML*, 2020.

[20] A. Roy, M. Saffar, A. Vaswani, and D. Grangier, "Efficient content-based sparse attention with routing transformers," in *Proc. TACL*, vol. 9, 2021, pp. 53–68.

[21] J. Kim, J. Lee, and Y. Lee, "Generalizing RNN-transducer to out-domain audio via sparse self-attention layers," in *Proc. INTERSPEECH*, 2022.

[22] S. Kim, S. Shen, D. Thorsley, A. Gholami, J. Hassoun, and K. Keutzer, "Learned token pruning for transformers," in *Proc. KDD*, 2022, pp. 784–794.

[23] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," in *Proc. NIPS*, vol. 33, 2020, pp. 12 449–12 460.

[24] 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 preprint arXiv:1904.02882*, 2019.

[25] J. Yamagishi, C. Veaux, K. MacDonald *et al.*, "Cstr vctk corpus: English multi-speaker corpus for cstr voice cloning toolkit (version 0.92)," 2019.

[26] J. Kong, J. Kim, and J. Bae, "Hifi-GAN: Generative adversarial networks for efficient and high fidelity speech synthesis," in *Proc. NIPS*, vol. 33, 2020, pp. 17 022–17 033.

[27] M. Ravanelli, T. Parcollet, P. Plantinga, A. Rouhe, S. Cornell, L. Lugosch, C. Subakan, N. Dawalatabad, A. Heba, J. Zhong *et al.*, "Speechbrain: A general-purpose speech toolkit," *arXiv preprint arXiv:2106.04624*, 2021.

[28] B. Desplanques, J. Thienpondt, and K. Demuynck, "ECAPA-TDNN: Emphasized channel attention, propagation and aggregation in TDNN based speaker verification," in *Proc. INTERSPEECH*, 2020, pp. 3830–3834.