INTERSPEECH 2023
20-24 August 2023, Dublin, Ireland

Hierarchical Timbre-Cadence Speaker Encoder for Zero-shot Speech Synthesis

Joun Yeop Lee, Jae-Sung Bae, Seongkyu Mun, Jihwan Lee, Ji-Hyun Lee, Hoon-Young Cho, Chanwoo Kim

Samsung Research, Seoul, South Korea

{jounyeop.lee, js3.bae, sk1213.mun, jihwan15.lee, jihyuny.lee, h.y.cho, chanw.com}@samsung.com

Abstract

Although recent zero-shot text-to-speech (zs-TTS) models have shown high performance in terms of speech quality, speaker similarity is not up to par. Speaker similarity can be expressed in two different components: intra-speaker consistent component (timbre) and inter-utterance variate component (cadence). In this paper, we propose a timbre-cadence speaker encoder for zs-TTS that improves speaker similarity by modeling these components. To disentangle timbre and cadence more efficiently, we employ a hierarchical structure. The cadence embedding is first encoded with VICReg which enlarges the inter-utterance embedding within a batch. Next, timbre embedding is extracted after subtracting cadence embedding and using a loss between timbre embedding and speaker ID-based speaker embedding. Additionally, we propose an effective data augmentation called speaker mixing augmentation, where two short utterances from different speakers are concatenated for a more robust zs-TTS model.

Index Terms: speech synthesis, zero-shot, speaker encoder, timbre, cadence

1. Introduction

In recent years, text-to-speech (TTS) has accomplished remarkable improvement with the emergence of various end-to-end TTS models [1, 2, 3]. Through these advanced models, TTS expands its field from a model built with a professional voice actor to a personalized TTS. To make the TTS model personalize, there have been several attempts to fine-tune the pre-trained model with a target speaker data [4, 5]. Even though the required quantity of the target speaker data for the fine-tuning is small, it is troublesome to collect such personal data and perform fine-tuning. The zero-shot TTS (zs-TTS) [6, 7, 8, 9, 10, 11], which uses a single utterance to clone the voice of the target speaker without additional fine-tuning, resolves these inconveniences. Nowadays, large language model-based zs-TTS [12, 13] has appeared and caught attention with its impressive quality. However, these kinds of models require massive data, and large resources to train the giant model. Instead, the more common approach in zs-TTS is to condition speaker embedding to typical end-to-end TTS architecture.

As perceiving speaker similarity is a complex and ambiguous problem, an insightful definition of *speaker similarity* should precede. Thus in this paper, we assume that the speaker similarity between two speeches can be viewed in two aspects. To clarify, we restrict the definition of two terms to explain these aspects. First, each speaker has a unique identity of voice regardless of the textual context of an utterance, and we will call this property as *timbre*. The other term is *cadence*, which is an inter-utterance variant component related to prosody, style, and variation of tone within the same speaker. For example, when someone has a severe hoarse throat, they lose their original timbre. Likewise, when two utterances are in different cadences, such as dialects, they are not usually considered to be from the same speaker. As timbre and cadence have highly complicated relations, it is important to design an adequate speaker encoder for zs-TTS architecture.

The most widely used speaker encoder method in the zs-TTS is a reference encoder-based approach. The reference encoder takes a reference speech sample as an input and output speaker embedding in a single vector [7, 14, 15] or sequence of fine-grained vectors [16, 17, 18]. These approaches typically do not regard the timbre and cadence separately but concentrate on extracting speaker characteristics or prosody in entangled form. Therefore, we cannot guarantee that the reference encoder model reflects the timbre or cadence sufficiently. Also, these approaches have limitations due to the lack of additional loss for modeling timbre and cadence separately. In [16], the authors suggested using a coarse and fine-grained encoder which is a similar concept with timbre and cadence, but they use separate encoders where timbre and cadence are not disentangled sufficiently.

Previous studies using external pre-trained speech encoder modules have also been proposed [6, 8, 19]. It transfers speaker information from a well-trained speaker verification (SV) model to the zs-TTS model. As the SV models are typically trained with a large number of speakers, they accomplish good speaker generalization. However, since the purpose of the SV model is to verify the identity of the speaker, it focuses on detecting timbre rather than cadence.

To model timbre and cadence separately, we propose a timbre-cadence speaker encoder (TiCa) that has a hierarchical structure. A hierarchically structured encoder has been proposed to capture and disentangle the global and local information [20, 21, 22]. However, these methods concentrate on modeling prosody rather than speaker identity. In addition to the hierarchical structure, to extract intra-speaker invariant timbre embedding, we constrain timbre embedding to be close to the embedding achieved from the speaker ID-based embedding table. We can adopt the speaker ID-based embedding table directly as timbre embedding, however, it is impossible to handle the zero-shot scenario. As the cadence embedding should have a large variance among the utterances, we employ VICReg [23] which provides regularization to enlarge the variance of cadence embedding within a batch.

In our preliminary experiments, the conventional zs-TTS models have shown unstable performance such as switching timbre within an utterance. Thus, we propose an effective augmentation method called speaker mixing augmentation (SMAug) to make a robust zs-TTS model. SMAug concatenates

10.21437/Interspeech.2023-1128

two different utterances from other speakers. By this, the model is exposed to the switching cases of speaker embeddings during training, which results in a robust model. Moreover, since SMAug makes various combinations of utterances, it augments the training dataset and reinforces the robustness.

2. Proposed method

2.1. Timbre-cadence speaker encoder

As mentioned in Section 1, timbre is an intra-speaker consistency component of a speaker that is globally unique regardless of what the speaker speaks. On the other hand, cadence is an inter-utterance variant component of a speaker that varies locally depending on the utterance. To separate these components, we suggest a hierarchical speaker encoder as in Figure 1. As the speaker encoder encodes gradually from local to global information, the cadence embedding is first extracted using the attention pooling at the lower part of the speaker encoder. For attention pooling, we use the method of deriving a weighted mean vector as attentive statistical pooling in [24]. Then, the output of the attention pooling layer of the cadence embedding is subtracted to disentangle the timbre and cadence information. The timbre embedding is extracted after two convolution blocks followed by attention pooling. Finally, the timbre and cadence embeddings are concatenated and form the speaker embedding conditioned on the zero-shot TTS (zs-TTS) framework.

Furthermore, we utilize two supplementary losses for training the timbre-cadence speaker encoder (TiCa). First, to keep the timbre embedding consistent within the same speaker, we give $l1$-loss L_{timb} against the speaker ID-based speaker embedding acquired by an embedding table. Before L_{timb}, we stop the gradient of the speaker ID-based speaker embedding, so that the training of the timbre embedding does not affect any other modules in the TTS model.

Second, we adopt the VICReg [23] to regularize the cadence embeddings within a batch to be distinct from each other and to decorrelate the variables of each cadence embedding. We denote a cadence embedding batch as $Z_{cad} = [z_1, ..., z_N]$, and the vector consists of each element in d-th dimension of z_n in Z_{cad} as z^d. The variance regularization term REG_{var} and the covariance regularization term REG_{cov} are as follows:

$$REG_{var} = \frac{1}{D}\sum_{d=1}^{D} \max\left(0, \gamma - \sqrt{Var(z^d + \epsilon)}\right), \quad (1)$$

$$REG_{cov} = \frac{1}{D}\sum_{i \neq j} \left(Cov(Z_{cad})\right)_{i,j}^2, \quad (2)$$

where D, Var, and Cov are the dimension of embedding, variance of vector z^d, and covariance matrix of Z_{cad}, respectively. Also, γ is a target of standard deviations, and ϵ is a small scalar value to prevent instability of the system. For our experiment, γ and ϵ are fixed to 1 and 10^{-4}.

By using such losses, we can successfully restrict the timbre embedding to have a smaller discrepancy within the same speaker and enlarge the diversity of the cadence embedding among different utterances. Finally, the total supplementary loss L_{sup} can be denoted as follows:

$$L_{sup} = L_{timb} + \lambda_v REG_{var} + \lambda_c REG_{cov}, \quad (3)$$

where λ_v and λ_c are the weight of regularization for REG_{var} and REG_{conv}, respectively. In the experiments, we fix these weights as $\lambda_v = \lambda_c = 3.0$. The L_{sup} is added to the other TTS losses to train the zs-TTS model.

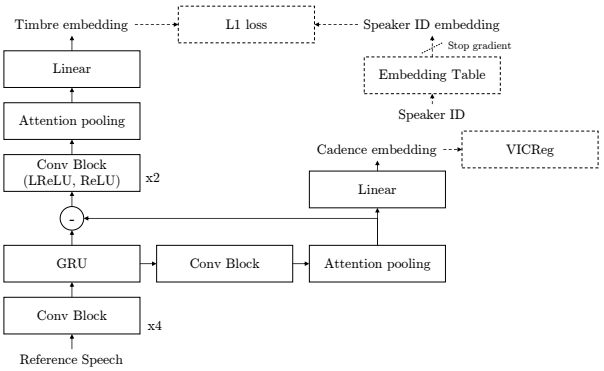


Figure 1: *The overall architecture of the TiCa. The dashed lines represent the components that are only used during training.* ***Conv Block*** *consists of a convolution layer, a normalization layer (default: batch normalization), and an activation layer (default: ReLU).*

To train the TiCa stably, in the early stage of the training, we use the speaker ID embedding instead of the timbre embedding. Then, to reduce the mismatch of the training and inference, we train the zs-TTS model conditioning on the timbre embedding while freezing the speaker ID embedding table for certain iterations. In our experiments, we trained the zs-TTS model for 450K steps with speaker ID embedding and 50K steps with timbre embedding, respectively. Note that our proposed methods can be used in the zero-shot scenario because we use the timbre embedding extracted from the reference speech instead of the speaker ID embedding.

2.2. Speaker mixing augmentation

In typical zs-TTS models, a single vector speaker embedding is broadcasted into the phoneme sequence length or acoustic feature sequence length and then used as a condition on TTS model. However, since the speaker embedding is fixed throughout the whole utterance, it can weaken the role of the speaker embedding and result in synthesized speech with unstable speaker similarity during inference.

To overcome this problem, SMAug concatenates two short utterances from different speakers. As long utterances slow the training, we constrain the candidate of SMAug to short utterances. We only adopt the SMAug to utterances that have lengths shorter than half of the longest utterance. Using SMAug, the model encounters two different speakers in one integrated utterance, which enhances the robustness of the zs-TTS model.

Data augmentation in TTS is limited since modification in target speech can result in severe performance degradation in naturalness. However, as the SMAug does not perturb the speech samples, it can augment the data while not harming the naturalness of the TTS models.

At every epoch, the SMAug performs as follows: First, for every utterance in a dataset, if the length of utterance is shorter than half of the max length of utterance in the data, we decide whether to do augmentation or not with a probability of p ($p = 0.5$ in our experiments). Second, if augmentation is decided, concatenate another short utterance, which is shorter than half of the max length, in distinct speakers.

Furthermore, as the SMAug has an advantage in compatibility, it can be expanded to other applications easily. For example, for expressive TTS scenarios, we can mix utterances

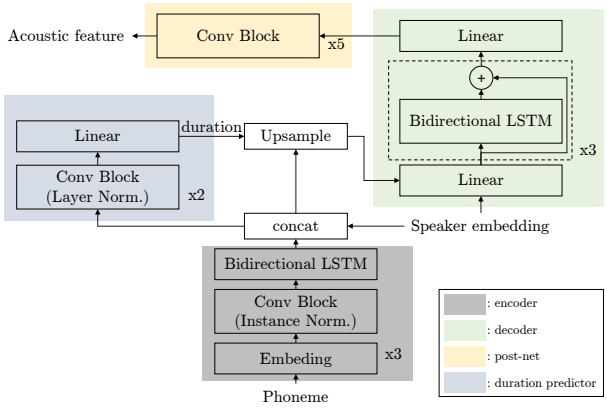


Figure 2: *The overall architecture of the NALPCTron.*

depending on the emotion instead of the speaker or even mix utterances depending on both emotion and speaker.

3. Experiments

3.1. Model

To show the performance of our proposed method, we performed experiments by replacing speaker encoder modules in a fixed end-to-end model which is a non-attentive version of [1] (NALPCTron). The overall architecture of an acoustic model in NALPCTron is in Figure 2 and we employed the Bunched LPCNet [25] as a neural vocoder. NALPCTron's encoder encodes phoneme sequence to phoneme embedding, and the duration predictor predicts the duration of each phoneme. Using duration, we upsample the phoneme embedding and feed it to the decoder with the speaker embedding to synthesize acoustic features. To predict the duration to be used during training, we adopted the alignment method used in [26].

3.2. Dataset

All comparison models were trained on the LibriTTS [27] train-clean-360 set. LibriTTS train-clean-360 set is a multi-speaker text-audio pair dataset that contains approximately 191 hours of speech with a 24 kHz sampling rate recorded by 904 speakers. We randomly selected 240 utterances from arbitrary 12 speakers as a seen test set, and the test set is excluded during training. To evaluate the performance on the zero-shot scenario, we used the small subset of LibriTTS test-clean set (approximately 8 hours of audio from 39 speakers with a 24 kHz sampling rate) with 12 speakers, and text from LJSpeech [28].

We used 22-dimensional acoustic features consisting of 20 Bark cepstral coefficients, pitch period, and pitch correlation which were the same as in [25] since we used the Bunched LPCNet. These acoustic features were used as the target of the acoustic model and the input of the speaker encoder.

3.3. Experiment setup

We evaluated the timbre-cadence speaker encoder (**TiCa**) with the conventional reference encoder-based speaker encoders. First, we utilized a vanilla reference encoder [14]-based speaker embedding (**REF**) which consists of convolution blocks and a recurrent pooling layer. Second, we applied Meta-StyleSpeech [15]-based speaker encoder (**META**) which is a self-attention-based method. Third, we adopted speaker embedding from the pre-trained speaker verification (SV) model[1] (**EXTERN**) [29]. Also, to show the effect of SMAug, we conduct some experiments without SMAug (**TiCa-NoAug**)[2]. To train each model, we utilized one NVIDIA A100 GPU device.

Table 1: *Objective experiment results on the seen case.*

Method	PER(%)	SECS
GT	1.77	N/A
TiCa-NoAug	2.85	0.38
TiCa	**1.79**	0.41
REF	2.58	0.41
META	2.53	0.40
EXTERN	2.25	**0.48**

Table 2: *Objective and subjective experiment results on the unseen case. MOS is represented with 95% confidence intervals.*

Method	PER	SECS	MOS	CSMOS
GT	2.85	N/A	4.48 ± 0.06	-
TiCa-NoAug	1.93	0.38	-	-
TiCa	**1.34**	0.42	$\mathbf{3.98 \pm 0.11}$	N/A
REF	1.76	0.34	3.82 ± 0.11	-0.192
META	2.33	0.32	3.58 ± 0.11	-0.139
EXTERN	1.88	**0.49**	3.59 ± 0.12	-0.145

3.4. Evaluation Metrics

For an objective test, we evaluated the phoneme error rate (PER) of the speech samples with an automatic speech recognition (ASR) model to predict pronunciation accuracy. Since the pronunciation unit of speech is phoneme, we used PER instead of word error rate or character error rate. We employ ASR model which is finetuned over the XLSR-53 model in Wav2vec 2.0 model [30] with the VCTK [31], LibriTTS (train-clean-100, train-clean-360), and LJSpeech [28] dataset.

In addition, averaged speaker embedding cosine similarity (SECS) was performed to evaluate the speaker similarity between the ground truth (**GT**) samples and synthesized samples. For the SECS, we adopted the pre-trained ECAPA-TDNN model [3] [32, 33] which is one of the SOTA SV models. The SECS ranges from 0 to 1, and a higher score implies better speaker similarity.

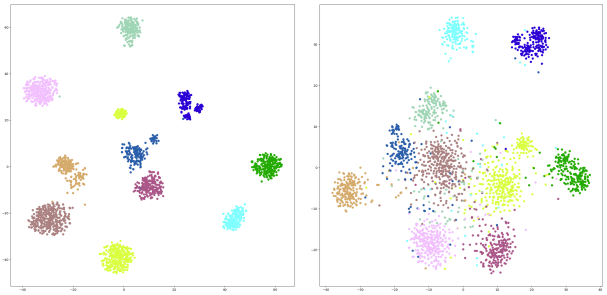
For the subjective evaluations, we measured the mean opinion score (MOS) and comparative similarity MOS (CSMOS). MOS estimates the perceptual speech quality by testers in the range from 1 to 5 with an interval of 1, where 5 is the best. For CSMOS, testers listened to one reference speech and two synthesized utterances including **TiCa**, then were asked to choose which utterance is similar to the reference speech in terms of timbre and cadence with ranges from -3 (the comparative model is much worse than **TiCa**) to 3 (the comparative model is much better than **TiCa**) with an interval of 1. For both subjective tests, 120 testers participated via Amazon MTurk[4].

[1] https://github.com/clovaai/voxceleb_trainer

[2] The samples of our experiments and details of model hyperparameters can be found in https://srtts.github.io/tc-zstts.

[3] https://github.com/TaoRuijie/ECAPA-TDNN

[4] https://www.mturk.com



(a) (b)

Figure 3: *The t-SNE visualization of the (a) timbre and (b) cadence embedding. The color of the dots represents the speaker.*

For the PER and MOS tests in the unseen cases, we randomly chose 12 speakers from LibriTTS test-clean set and used 20 and 10 sentences for the PER and MOS tests, respectively. To test speaker similarity, it is important to evaluate the speaker similarity with diverse reference speeches regardless of the text. Therefore, for the SECS and CSMOS tests, n_t sentences from the LJSpeech dataset were randomly selected, where $n_t = 10$ for SECS and 5 for CSMOS. As the reference speech for each test sentence, we applied n_s different speech samples that were randomly selected from every 12 speakers in the LibriTTS test-clean set, where $n_s = 20$ for SECS and 10 for CSMOS.

3.5. Evaluation on seen speakers

In the seen speaker case, **TiCa** showed the best performance in PER as shown in Table 1, which indicates that **TiCa** produces speech with high intelligibility. Also, since **Tica** showed better performance in all objective tests than **TiCa-NoAug**, we can verify that the SMAug is effective in improving pronunciation accuracy and speaker similarity of the zs-TTS model. Among the reference encoder-based methods (**REF**, **META**), our approach showed on-par quality in SECS, which implies that **TiCa** perform better or similar speaker similarity on the seen cases. However, **EXTERN** showed the best scores in SECS. Here, note that the SV evaluation system used for the SECS tests is specialized in comparing timbre. Moreover, as the **EXTERN** used the external-SV model, it can synthesize speech with higher speaker similarity with respect to timbre, but it cannot guarantee the higher speaker similarity in terms of cadence.

3.6. Evaluation on unseen speakers (zero-shot TTS)

In the zero-shot scenario, as shown in Table 2, **TiCa** showed superior performance in speech quality such as pronunciation accuracy and naturalness in terms of PER and MOS results. The PER of **GT** was worse than the others since some noisy speeches were contained in the **GT** while the synthesized speeches of the zs-TTS models were typically clean. However, when it comes to the MOS test, **GT** showed a higher score, because the testers were guided to focus on the perceptual naturalness of speech.

In terms of speaker similarity, the CSMOS results demonstrated that **TiCa** had better perceptual speaker similarity than the comparison TTS models. This was because **TiCa** models both the timbre and cadence of speakers. However, the overall tendency of SECS results was similar to the seen case. As mentioned in Section 1, as the human sense of speaker similarity is a combination of timbre and cadence, this can be a reason that the tendency of CSMOS and SECS was different. Especially, for **EXTERN**, it achieved the best SECS results but showed worse results than **TiCa** and **META** in the CSMOS test. This implies that **EXTERN** only focused on encoding the timbre of the speaker.

Table 3: *Ablation studies results on supplementary losses in the unseen case.*

Method	PER(%)	SECS
TiCa	**1.34**	**0.42**
TiCa-NoID	1.48	0.32
Tica-NoVICReg	1.48	0.40

Comparing the objective results of **TiCa** and **TiCa-NoAug**, it demonstrates a similar trend for the seen case, which proves the positive effect of SMAug in terms of pronunciation accuracy and speaker similarity.

To further verify whether the timbre and cadence are encoded well as intended, we visualize the t-SNE [34] of timbre and cadence embedding of **TiCa** as in Figure 3. The embeddings were extracted from 10 unseen speakers in LibriTTS test-clean set. Speakers are distinguished by different colors and each dot indicates different utterances. From Figure 3a, we can confirm that the timbre embedding is well clustered for each speaker. On the other hand, Figure 3b illustrates that the cadence embeddings are more scattered and have higher variation within inter-utterances even if they are from the same speaker.

3.7. Ablation study on supplementary losses

To study the effect of the supplementary losses in Section 2.1, we respectively turn off the L_{timb} and VICReg losses (REG_{var}, REG_{cov}) which are denoted as **TiCa-NoID** and **TiCa-NoVICReg**. **TiCa-NoID** and **TiCa-NoVICReg** showed worse performance than **TiCa** for all objective tests, which implies that using both losses is effective. **TiCa-NoID** achieved much lower SECS results than **TiCa** while that of **TiCa-NoVICReg** was slightly lower than **TiCa**. As the SECS focuses on the timbre information, we can conclude that the L_{timb} is important for the speaker encoder to well model the timbre information.

4. Conclusions

In this paper, we proposed a timbre-cadence speaker encoder (TiCa) as a novel technique for cloning a target speaker's voice. The proposed approach assumes that speaker embedding can be viewed as a combination of timbre and cadence. To model these components, TiCa extracts timbre and cadence with a hierarchical structure and some effective additional losses. Moreover, we proposed a simple but powerful speaker mixing augmentation (SMAug) that concatenates two utterances from the different speakers for robust zero-shot TTS. From the experimental results, it showed that our proposed methods outperform the conventional other reference encoder-based speaker encoders.

For future work, we plan to build a more suitable TiCa architecture by utilizing auxiliary linguistic information and to apply TiCa to other end-to-end TTS frameworks to broaden our method. We also aim to expand SMAug to other applications such as expressive TTS.

5. References

[1] N. Ellinas, G. Vamvoukakis, K. Markopoulos, A. Chalamandaris, G. Maniati, P. Kakoulidis, S. Raptis, J. S. Sung, H. Park, and P. Tsiakoulis, "High Quality Streaming Speech Synthesis with Low, Sentence-Length-Independent Latency," in *Proc. Interspeech*, 2020, pp. 2022–2026.

[2] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerrv-Ryan, R. A. Saurous, Y. Agiomvrgiannakis, and Y. Wu, "Natural TTS Synthesis by Conditioning WaveNet on Mel Spectrogram Predictions," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 4779–4783.

[3] 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 (ICLR)*, 2021.

[4] Y. Chen, Y. M. Assael, B. Shillingford, D. Budden, S. E. Reed, H. Zen, Q. Wang, L. C. Cobo, A. Trask, B. Laurie, Çaglar Gülçehre, A. V. D. Oord, O. Vinyals, and N. de Freitas, "Sample Efficient Adaptive Text-to-Speech," in *International Conference on Learning Representations (ICLR)*, 2019.

[5] M. Chen, X. Tan, B. Li, Y. Liu, T. Qin, sheng zhao, and T.-Y. Liu, "AdaSpeech: Adaptive Text to Speech for Custom Voice," in *International Conference on Learning Representations (ICLR)*, 2021.

[6] E. Casanova, J. Weber, C. D. Shulby, A. C. Junior, E. Gölge, and M. A. Ponti, "YourTTS: Towards Zero-Shot Multi-Speaker TTS and Zero-Shot Voice Conversion for Everyone," in *International Conference on Machine Learning (ICML)*, 2022, pp. 2709–2720.

[7] 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, pp. 788–792.

[8] E. Casanova, C. Shulby, E. Gölge, N. M. Müller, F. S. de Oliveira, A. Candido Jr., A. da Silva Soares, S. M. Aluisio, and M. A. Ponti, "SC-GlowTTS: An Efficient Zero-Shot Multi-Speaker Text-To-Speech Model," in *Proc. Interspeech*, 2021, pp. 3645–3649.

[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, pp. 2568–2572.

[10] J.-H. Lee, S.-H. Lee, J.-H. Kim, and S.-W. Lee, "PVAE-TTS: Adaptive Text-to-Speech via Progressive Style Adaptation," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 6312–6316.

[11] B. J. Choi, M. Jeong, J. Y. Lee, and N. S. Kim, "SNAC: Speaker-Normalized Affine Coupling Layer in Flow-Based Architecture for Zero-Shot Multi-Speaker Text-to-Speech," *IEEE Signal Processing Letters*, vol. 29, pp. 2502–2506, 2022.

[12] C. Wang, S. Chen, Y. Wu, Z. Zhang, L. Zhou, S. Liu, Z. Chen, Y. Liu, H. Wang, J. Li, L. He, S. Zhao, and F. Wei, "Neural Codec Language Models are Zero-Shot Text to Speech Synthesizers," *arXiv:2301.02111*, 2023.

[13] E. Kharitonov, D. Vincent, Z. Borsos, R. Marinier, S. Girgin, O. Pietquin, M. Sharifi, M. Tagliasacchi, and N. Zeghidour, "Speak, Read and Prompt: High-Fidelity Text-to-Speech with Minimal Supervision," *arXiv:2302.03540*, 2023.

[14] R. J. Skerry-Ryan, E. Battenberg, Y. Xiao, Y. Wang, D. Stanton, J. Shor, R. J. 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 (ICML)*, 2018, pp. 4700–4709.

[15] D. Min, D. B. Lee, E. Yang, and S. J. Hwang, "Meta-StyleSpeech: Multi-Speaker Adaptive Text-to-Speech Generation," in *International Conference on Machine Learning (ICML)*, 2021, pp. 7748–7759.

[16] S. Choi, S. Han, D. Kim, and S. Ha, "Attentron: Few-Shot Text-to-Speech Utilizing Attention-Based Variable-Length Embedding," in *Proc. Interspeech*, 2020, pp. 2007–2011.

[17] D. Tan and T. Lee, "Fine-Grained Style Modeling, Transfer and Prediction in Text-to-Speech Synthesis via Phone-Level Content-Style Disentanglement," in *Proc. Interspeech*, 2021, pp. 4683–4687.

[18] Y. Lee and T. Kim, "Robust and fine-grained prosody control of end-to-end speech synthesis," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 5911–5915.

[19] Y. Jia, Y. Zhang, R. Weiss, Q. Wang, J. Shen, F. Ren, z. Chen, P. Nguyen, R. Pang, I. Lopez Moreno, and Y. Wu, "Transfer Learning from Speaker Verification to Multispeaker Text-To-Speech Synthesis," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

[20] C. Chien and H. Lee, "Hierarchical Prosody Modeling for Non-Autoregressive Speech Synthesis," in *IEEE Spoken Language Technology Workshop (SLT)*, 2021, pp. 446–453.

[21] J.-S. Bae, J. Yang, T. Bak, and Y.-S. Joo, "Hierarchical and Multi-Scale Variational Autoencoder for Diverse and Natural Non-Autoregressive Text-to-Speech," in *Proc. Interspeech*, 2022, pp. 813–817.

[22] J.-S. Bae, T. Bak, Y.-S. Joo, and H.-Y. Cho, "Hierarchical Context-Aware Teransformers for Non-Autoregressive Text to Speech," in *Proc. Interspeech*, 2021, pp. 3610–3614.

[23] A. Bardes, J. Ponce, and Y. LeCun, "VICReg: Variance-Invariance-Covariance Regularization for Self-Supervised Learning," in *International Conference on Learning Representations (ICLR)*, 2022.

[24] K. Okabe, T. Koshinaka, and K. Shinoda, "Attentive Statistics Pooling for Deep Speaker Embedding," in *Proc. Interspeech*, 2018, pp. 2252–2256.

[25] S. Park, K. Choo, J. Lee, A. V. Porov, K. Osipov, and J. S. Sung, "Bunched LPCNet2: Efficient Neural Vocoders Covering Devices from Cloud to Edge," in *Proc. Interspeech*, 2022, pp. 808–812.

[26] R. Badlani, A. Łańcucki, K. J. Shih, R. Valle, W. Ping, and B. Catanzaro, "One TTS Alignment to Rule Them All," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 6092–6096.

[27] 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," in *Proc. Interspeech*, 2019, pp. 1526–1530.

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

[29] J. S. Chung, J. Huh, S. Mun, M. Lee, H. S. Heo, S. Choe, C. Ham, S. Jung, B.-J. Lee, and I. Han, "In Defence of Metric Learning for Speaker Recognition," in *Proc. Interspeech*, 2020.

[30] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2020, pp. 12 449–12 460.

[31] J. Yamagishi, C. Veaux, and K. MacDonald, "CSTR VCTK Corpus: English Multi-speaker Corpus for CSTR Voice Cloning Toolkit (version 0.92)," 2019.

[32] 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.

[33] R. K. Das, R. Tao, and H. Li, "HLT-NUS SUBMISSION FOR 2020 NIST Conversational Telephone Speech SRE," *arXiv:2111.06671*, 2021.

[34] L. van der Maaten and G. Hinton, "Visualizing Data using t-SNE," *Journal of Machine Learning Research*, vol. 9, no. 86, pp. 2579–2605, 2008.