ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 979-8-3503-6874-1/25/$31.00 ©2025 IEEE | DOI: 10.1109/ICASSP49660.2025.10889369

NanoVoice: Efficient Speaker-Adaptive Text-to-Speech for Multiple Speakers

Nohil Park
Electrical and Computer Engineering
Seoul National University
Seoul, Republic of Korea
pnoil2588@snu.ac.kr

Heeseung Kim
Electrical and Computer Engineering
Seoul National University
Seoul, Republic of Korea
gmltmd789@snu.ac.kr

Che Hyun Lee
Electrical and Computer Engineering
Seoul National University
Seoul, Republic of Korea
saga1214@snu.ac.kr

Jooyoung Choi
Electrical and Computer Engineering
Seoul National University
Seoul, Republic of Korea
jy_choi@snu.ac.kr

Jiheum Yeom
Electrical and Computer Engineering
Seoul National University
Seoul, Republic of Korea
quilava1234@snu.ac.kr

Sungroh Yoon*
ECE, AIIS, ASRI, INMC, ISRC, and IPAI
Seoul National University
Seoul, Republic of Korea
sryoon@snu.ac.kr

***Abstract*—We present NanoVoice, a personalized text-to-speech model that efficiently constructs voice adapters for multiple speakers simultaneously. NanoVoice introduces a batch-wise speaker adaptation technique capable of fine-tuning multiple references in parallel, significantly reducing training time. Beyond building separate adapters for each speaker, we also propose a parameter sharing technique that reduces the number of parameters used for speaker adaptation. By incorporating a novel trainable scale matrix, NanoVoice mitigates potential performance degradation during parameter sharing. NanoVoice achieves performance comparable to the baselines, while training 4 times faster and using 45 percent fewer parameters for speaker adaptation with 40 reference voices. Extensive ablation studies and analysis further validate the efficiency of our model.**

***Index Terms*—text-to-speech, TTS, speaker adaptation, multiple speakers, parameter-efficient TTS**

I. Introduction

With the advancement of text-to-speech (TTS) method [1], [2], various speaker-adaptive TTS models [3]–[5] have been introduced to accurately mimic the target speaker's voice. Speaker-adaptive TTS methods are primarily categorized into zero-shot and one-shot approaches. The zero-shot approach [3]–[7], which does not incur additional training costs for adaptation, necessitates a large dataset and numerous parameters to construct a TTS model and often struggles with unique out-of-distribution (OoD) voices. In contrast, the one-shot approach [8]–[12] necessitates fine-tuning of a pre-trained multi-speaker TTS model but effectively adapts to the desired speaker's voice. This fine-tuning based approach not only enhances robustness against OoD data but also reduces the data and model size requirements during the pre-training phase.

* Corresponding Author

This work was supported by Samsung Electronics Co., Ltd (IO231120-07949-01) and Korean Government (2022R1A3B1077720, 2022R1A5A708390811, RS-2022-II220959, BK21 Four Program, IITP-2024-RS-2024-00397085 & RS-2021-II211343: AI Graduate School Program)

Recently, leveraging the capabilities of diffusion-based generative models, various diffusion-based personalization models have been proposed across diverse applications such as text-to-image [13], demonstrating the ability to achieve personalization with minimal data. This trend extends to one-shot TTS [14], [15] as well, where adaptation is enabled by fine-tuning the entire parameters of pre-trained diffusion-based TTS models with just 5-10 seconds of target reference speech. More recent efforts have incorporated parameter-efficient fine-tuning techniques, such as low-rank adaptation (LoRA) [16], with one-shot TTS [17], to efficiently perform speaker adaptation with higher speaker similarity.

Including the aforementioned research on personalization, previous works on fine-tuning pre-trained models have predominantly focused on fine-tuning single tasks [18], [19] or single reference samples [13]–[15]. Recently, the commercialization of various deep learning models has intensified the need to handle multiple queries in parallel [20], [21]. This demand is even more critical for fine-tuning methods, as a naive approach would be to fine-tune for each task sequentially, which is computationally inefficient and memory-intensive. As a result, research into more efficient fine-tuning methodologies has been propelled forward, where a single fine-tuning process aims to address multiple tasks [22] or perform personalization using several reference samples simultaneously [23], [24]. Although such approaches have improved the efficiency of handling multiple queries, their applications in one-shot TTS remain unexplored.

In this work, we propose NanoVoice, an efficient speaker-adaptive TTS designed to perform adaptations for multiple voices simultaneously. Utilizing VoiceTailor [17], a one-shot TTS that employs LoRA for fine-tuning, as its backbone, NanoVoice accelerates speaker adaptation for multiple references through introducing a batch-wise fine-tuning scheme. Additionally, instead of constructing separate adapters for each of the references, NanoVoice shares parts of the adapters

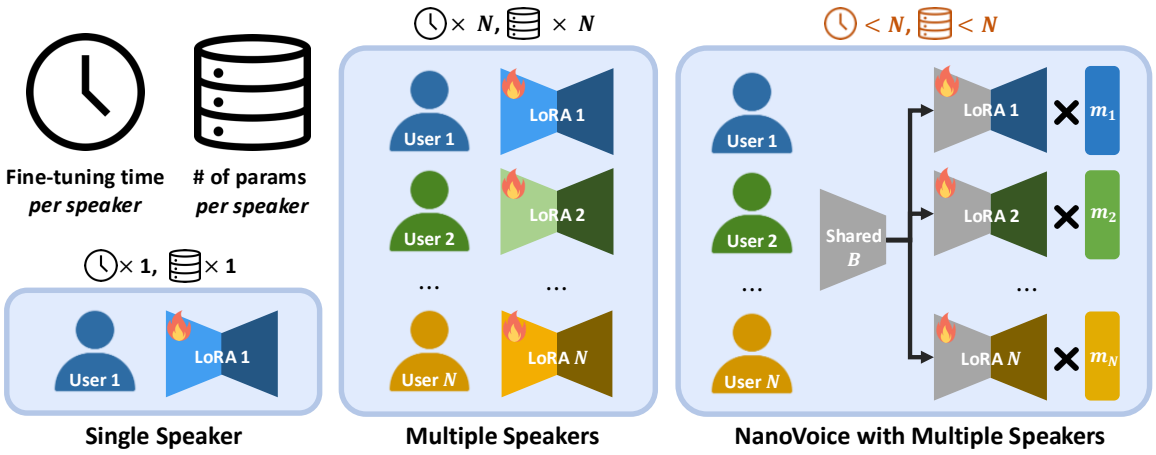

Fig. 1. An overview of speaker adaptation in various scenarios: single reference adaptation, sequential adaptation of multiple references, and NanoVoice.

across all references for parameter efficiency. Enhanced by an additional trainable scale matrix, our method enables NanoVoice to adapt to multiple voices efficiently in terms of both parameters and fine-tuning time.

We demonstrate that NanoVoice, when fine-tuned with 40 references for adaptation in parallel, achieves performance comparable to one-shot baselines while being 4 times faster and using 45% fewer parameters. Moreover, NanoVoice exhibits comparable or superior performance to zero-shot baselines, despite the latter using significantly more data for pre-training. Our ablation studies validate the effectiveness of each component of NanoVoice. Additionally, we conduct several experiments to explore the characteristics and robustness of NanoVoice. Audio samples are on our demo page[1].

II. Method

We introduce NanoVoice, a model that efficiently personalizes multiple reference audios simultaneously. Fig. 1. compares NanoVoice with sequential fine-tuning. NanoVoice builds upon VoiceTailor [17], a parameter-efficient one-shot TTS model that advances beyond UnitSpeech [15], which fine-tunes all parameters, by integrating the LoRA [16] for parameter-efficient one-shot TTS (Section II-A). NanoVoice improves both time and parameter efficiency via batch operations and parameter sharing (Section II-B), with a lightweight scale matrix preventing performance loss (Section II-C).

A. UnitSpeech and VoiceTailor

UnitSpeech [15], a diffusion-based [25] one-shot TTS model, first defines the forward process that progressively transforms the mel-spectrogram X_0 into noise vector $X_1 \sim N(0, I)$. Given a noise schedule β_t and a random noise vector $\epsilon_t \sim N(0, I)$, we can obtain the corrupted mel-spectrogram X_t at any timestep t as follows:

$$X_t = \sqrt{\lambda_t} X_0 + \sqrt{1-\lambda_t}\epsilon_t, \quad \lambda_t = \mathrm{e}^{-\int_0^t \beta_s ds}. \tag{1}$$

To synthesize speech with a voice of a target speaker S and a transcript c, the reverse trajectory of the pre-defined forward process is required, and the score $\nabla_{X_t} \log p(X_t|c, S)$ for the corrupted mel-spectrogram X_t is needed. Therefore, UnitSpeech trains a network $s_\theta(X_t|c, S)$ to predict this score, which is then used for mel-spectrogram generation. The loss function used for training and the equation for generation involving the score network s_θ are as follows:

$$L(\theta) = \mathbb{E}_{t, X_0, \epsilon_t}[\| \sqrt{1-\lambda_t} s_\theta(X_t|c, S) + \epsilon_t \|_2^2], \tag{2}$$

$$X_{t-\Delta t} = X_t + \beta_t (\frac{1}{2} X_t + s_\theta(X_t|c, S))\Delta t + \sqrt{\beta_t \Delta t} z_t, \tag{3}$$

where z_t is a random vector that follows standard normal distribution $N(0, I)$.

UnitSpeech is pre-trained with (2) on LibriTTS [26], a large-scale multi-speaker TTS dataset to predict scores for multiple speakers following [1]. During fine-tuning, it adapts all of its parameters using the reference data of the target speaker. Unlike UnitSpeech, VoiceTailor [17] achieves parameter-efficient speaker adaptation by injecting a low-rank adapter into the linear layers of the attention modules in the pre-trained TTS model and fine-tuning only the adapter parameters. Specifically, for a linear layer with matrix W_0, VoiceTailor injects a matrix $\Delta W = BA$ where $B \in \mathbb{R}^{d \times r}$ and $A \in \mathbb{R}^{r \times k}$ with a scale factor α, resulting in $W = W_0 + \alpha \cdot BA$. By setting r much smaller than d and k, the number of parameters in B and A becomes significantly smaller than in W_0, allowing for fine-tuning with far fewer parameters. In this paper, we follow the observation that $r = 2$ is sufficient for effective speaker adaptation in [17], and therefore set r of NanoVoice to 2.

B. Batch-Wise Fine-tuning Scheme with Parameter Sharing

NanoVoice extends VoiceTailor by building multiple adapters simultaneously using multiple reference voices. Given N reference voices, we first batch the N reference speeches to create a batched reference sample $X_0' \in \mathbb{R}^{N \times L}$, where L is the maximum length of the mel-spectrograms of the reference samples. We then stack N low-rank matrices for each reference to form new matrices $B' \in \mathbb{R}^{b \times d \times r}$ and $A' \in \mathbb{R}^{b \times r \times k}$. During fine-tuning, batch-wise matrix multiplication ensures that the loss and gradient for each reference sample

[1]Demo: https://nanovoice.github.io/

are calculated separately. This approach, performing independent computations within the batch, maintains performance comparable to the method of constructing multiple adapters sequentially while achieving faster speaker adaptation.

In pursuit of further efficiency, we propose to share parameters that are less critical for personalization across multiple voices. To investigate which parameters to share, we run an experiment by selecting 50 random samples from the `test-clean` subset of LibriTTS [26] as reference data. Using this, we perform speaker adaptation with four configurations of low-rank adapters: the baseline setup with both matrices batch-wise (B', A'), a setup where B is shared across all references (B, A'), a setup where A is shared across all references (B', A), and a setup where both matrices are shared (B, A). For each, we build personalized speech models and measure the speaker similarity between the generated and the reference speech, as described in Section III-A.

The results show that sharing B led to the least performance drop, with SECS values of 0.938 for the batch-wise setup, 0.933 for sharing B, 0.902 for sharing A, and 0.898 for sharing both B and A. Therefore, NanoVoice utilizes A' in a batch-wise manner and shares B across all reference voices. Given that the number of parameters for B' constitutes approximately two-thirds of the total trainable parameters, this sharing approach allowed us to reduce the number of parameters by nearly threefold.

C. Lightweight Scale Matrix

While the proposed method with batch-wise matrix A' combined with a shared matrix B reduces the number of trainable parameters, it introduces a slight performance degradation, as seen in the previous section. To mitigate this, inspired by DoRA [27], one of several methods for boosting the capacity of LoRA with a minimal increase in parameters [27]–[29], we introduce a trainable scale matrix $m' \in \mathbb{R}^{N\times 1\times k}$, composed of stacked scale vectors for each reference.

The scale matrix m' is initialized with the column-wise weight norm of the pre-trained model W_0. To compute W during fine-tuning and inference, rather than directly applying m' to $W_0 + \alpha \cdot BA'$, we first normalize $W_0 + \alpha \cdot BA'$ by its column-wise norm, denoted as $||W_0 + \alpha \cdot BA'||_c$. Subsequently, we perform an element-wise multiplication with the scale matrix m', similar to [27]. Note that m' is batched across multiple speaker references, in contrast to training a single scale vector [27]. Although this method involves fewer additional parameters compared to the original low-rank adapter, it effectively enhances performance, making it a parameter-efficient solution for improving speaker adaptation.

III. Experiments

A. Experimental Setup

Datasets. We use the LibriTTS dataset [26], which contains 2,456 speakers, to train the pre-trained TTS model, following the same method as in UnitSpeech and VoiceTailor. For fine-tuning and evaluation, we utilize the `test-clean` subset of LibriSpeech, consisting of 40 speakers (20 male and 20 female). For each speaker, one reference audio is used for fine-tuning and one transcript is used for generation.

Pre-training and Fine-tuning Details. We follow VoiceTailor's procedure for pre-training a multi-speaker TTS model. For fine-tuning, we train the LoRA weights and scale matrices for 500 iterations using the Adam optimizer [32] with a learning rate of 0.0001. We set the LoRA rank to 2 and the scaling factor α to 8. This results in 21,363 trainable parameters *per speaker* for NanoVoice. Single-speaker adaptation takes approximately 7.6 seconds on a single NVIDIA A40 GPU.

Evaluations. NanoVoice, a one-shot TTS model capable of fine-tuning multiple references in parallel, uses adaptation models UnitSpeech [15] and VoiceTailor as baselines. We also use zero-shot TTS models, XTTS *v2* [30] and CosyVoice [31] for comparison. NanoVoice utilizes the official BigVGAN checkpoint [33] as its vocoder, with sampling procedure following the method used in VoiceTailor.

We perform both qualitative and quantitative evaluations using 40 sentences from the LibriSpeech `test-clean` subset. We use MTurk to measure 5-scale mean opinion score (MOS) for evaluating both audio quality and naturalness, as well as 5-scale speaker similarity mean opinion score (SMOS) for assessing speaker similarity. Additionally, speaker encoder cosine similarity (SECS) is measured using Resemblyzer speaker encoder [34], and pronunciation accuracy is assessed with character error rate (CER) using CTC-based Conformer [35] of NEMO toolkit [36]. All samples are generated five times using different seeds for fair comparison, and the average SECS and CER values are reported.

B. Model Comparison

As shown in Table I, NanoVoice shows comparable MOS and SMOS to other one-shot TTS baselines while using fewer trainable parameters *per speaker*. Notably, NanoVoice achieves speaker adaptation with just 21K trainable parameters, which is less than 0.02% of the full fine-tuning requirements of UnitSpeech and approximately 53.8% of the LoRA-based adaptation used by VoiceTailor. Additionally, NanoVoice's capacity to adapt to multiple speakers simultaneously enables a training speed 4 times faster than that of baseline models.

We also compare NanoVoice with the zero-shot TTS models. XTTS *v2*, which uses approximately 50 times the amount of NanoVoice's pre-training data, shows degraded SMOS performance compared to NanoVoice ($p < 0.05$). As demonstrated in the comparison with CosyVoice, zero-shot TTS models require nearly 300 times larger dataset to achieve similar levels of quality, naturalness, and speaker similarity as NanoVoice, which is reflected in the MOS and SMOS scores.

C. Ablation Studies

In Table II, the ablations on the LibriSpeech `test-clean` subset reaffirm the results of the experiments discussed in Section II-B. Notably, sharing B across all references shows comparable SECS while using only 37.1% of the trainable parameters compared to the batch-wise adapter setup, without

TABLE I

THE RESULTS OF 5-SCALE MOS, CER, 5-SCALE SMOS FOR ONE-/ZERO-SHOT TTS MODELS TESTED ON LIBRISPEECH. **# PARAMS** AND **ADAPTATION TIME** REFER TO THE NUMBER OF TRAINABLE PARAMETERS PER SPEAKER AND THE ADAPTATION TIME PER SPEAKER, RESPECTIVELY.

Method	Amount of Dataset	Fine-tuning	# Params	Adaptation Time	5-scale MOS	CER (%)	5-scale SMOS
Ground Truth	–	–	–	–	4.20 ± 0.08	0.82%	3.89 ± 0.07
NanoVoice	$\approx$ 585 hrs	✓	$21K$	$7.6s$	4.10 ± 0.09	1.10%	3.88 ± 0.08
VoiceTailor [17]	$\approx$ 585 hrs	✓	$39K$	$31s$	4.01 ± 0.10	1.17%	3.84 ± 0.09
UnitSpeech [15]	$\approx$ 585 hrs	✓	$119M$	$32s$	4.06 ± 0.09	1.14%	3.85 ± 0.09
XTTS *v2* [30]	27,281 hrs	✗	0	$0s$	4.00 ± 0.09	1.26%	3.74 ± 0.09
CosyVoice [31]	171,800 hrs	✗	0	$0s$	4.14 ± 0.09	3.05%	3.86 ± 0.08

TABLE II

ABLATION STUDIES FOR ADAPTER SHARING. **# PARAMS** REFERS TO THE NUMBER OF TRAINABLE PARAMETERS PER SPEAKER.

Share B	Share A	# Params	CER (%)	SECS
✗	✗	38,920	1.26%	0.870
✓	✗	14,459	1.23%	0.868
✗	✓	25,442	1.18%	0.815
✓	✓	973	1.13%	0.813

TABLE III

ABLATION STUDIES FOR THE TRAINABLE SCALE MATRIX. **# PARAMS** REFERS TO THE NUMBER OF TRAINABLE PARAMETERS PER SPEAKER.

Methods	# Params	CER (%)	SECS
NanoVoice	21,363	1.10%	0.873
– Normalization	21,363	1.47%	0.865
– Scale Matrix	14,459	1.23%	0.868

sharing any matrices. In contrast, sharing A results in a 1.75-fold increase in trainable parameters but leads to a decrease in performance compared to sharing B. The most significant performance drop occurs when both adapters are shared, implying that a single LoRA for all speakers is less effective.

In Table III, we present ablation experiments on the trainable scale matrix and operations proposed in Section II-C. The most intuitive approach to improve the training capacity of the shared matrix B with batched A' is to directly multiply m' without normalizing with $||W_0 + \alpha \cdot BA'||_c$. However, the results show that multiplying only the scale matrix slightly degrades SECS. This suggests that NanoVoice can enhance speaker similarity by ensuring training stability through the combined use of the normalization term.

D. Analysis

Number of Speakers. In our experiments, we train NanoVoice on the LibriSpeech `test-clean` subset by batching one reference audio per speaker and matching the number of voice adapters to the batch size. To confirm the robustness of NanoVoice's speaker similarity performance, we conduct an experiment with varying batch sizes. Since NanoVoice trains adapter groups for multiple speakers, Table IV shows that as the number of speakers trained simultaneously increases, parameter efficiency improves. Moreover, we observe that NanoVoice maintains consistent speaker similarity even when the number of batched adapters varies.

Role of Shared Matrix *B*. An important observation from Section II-B and Table II is that sharing the matrix B is sufficient for multi-speaker adaptation. We hypothesize that this observation is due to either B effectively models common information within the reference audio group, or B is not critical for speaker adaptation. To test these hypotheses, we conduct two analyses.

TABLE IV

ANALYSIS ON THE NUMBER OF SPEAKERS AND THE ROLE OF SHARED MATRIX B IN TERMS OF GENDER ATTRIBUTE.

Gender	Batch Size	# Params	CER (%)	SECS
–	1	45,832	1.30%	0.873
Mix	5	25,755	1.07%	0.872
Mix	20	21,991	1.18%	0.872
Mix	40	21,363	1.10%	0.873
Same	20	21,991	1.02%	0.872

First, to examine whether sharing B benefits the model, we analyze its ability to model gender, a distinct feature in speech. Using LibriSpeech's metadata, we create subsets with a batch size of 20, organized into *same*-gender and *mixed*-gender groups. The *same*-gender groups include two batches, either of 20 male or 20 female speakers, while the *mixed*-gender groups contain two batches with 10 male and 10 female speakers each. The results in Table IV show that training NanoVoice on each group yields similar performance, indicating that B does not capture common information within references, thus contradicting the first hypothesis.

Next, to test whether B is crucial for speaker adaptation, we freeze B and fine-tune only the remaining trainable parameters of NanoVoice. With the frozen matrix B, NanoVoice achieves SECS of 0.871, only 0.002 points lower than the original NanoVoice (0.873). This suggests that matrix B is not essential for multi-speaker adaptation, aligning with recent findings in the NLP field [37]. Therefore, we conclude that the effectiveness of parameter-efficient sharing in NanoVoice arises from the fact that B is not critical for speaker adaptation.

IV. CONCLUSION

In this work, we introduced NanoVoice, a parameter-efficient speaker-adaptive TTS method capable of handling multiple reference voices simultaneously. By employing a batch-wise training scheme and parameter sharing, along with learnable scale matrices, we have significantly reduced both the number of parameters and adaptation time per speaker. We hope that NanoVoice will pave the way for new opportunities in the commercialization of personalized TTS systems.

REFERENCES

[1] V. Popov, I. Vovk, V. Gogoryan, T. Sadekova, and M. Kudinov, "Grad-tts: A diffusion probabilistic model for text-to-speech," in *Proceedings of the 38th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, M. Meila and T. Zhang, Eds., vol. 139. PMLR, 18–24 Jul 2021, pp. 8599–8608.

[2] J. Kim, J. Kong, and J. Son, "Conditional variational autoencoder with adversarial learning for end-to-end text-to-speech," in *Proceedings of the 38th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, M. Meila and T. Zhang, Eds., vol. 139. PMLR, 18–24 Jul 2021, pp. 5530–5540.

[3] C. Wang, S. Chen, Y. Wu, Z.-H. 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*, vol. abs/2301.02111, 2023.

[4] M. Le, A. Vyas, B. Shi, B. Karrer, L. Sari, R. Moritz, M. Williamson, V. Manohar, Y. Adi, J. Mahadeokar, and W.-N. Hsu, "Voicebox: Text-guided multilingual universal speech generation at scale," in *Advances in Neural Information Processing Systems*, A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, Eds., vol. 36. Curran Associates, Inc., 2023, pp. 14 005–14 034.

[5] K. Shen, Z. Ju, X. Tan, E. Liu, Y. Leng, L. He, T. Qin, sheng zhao, and J. Bian, "Naturalspeech 2: Latent diffusion models are natural and zero-shot speech and singing synthesizers," in *The Twelfth International Conference on Learning Representations*, 2024.

[6] S. Kim, K. J. Shih, R. Badlani, J. F. Santos, E. Bakhturina, M. T. Desta, R. Valle, S. Yoon, and B. Catanzaro, "P-flow: A fast and data-efficient zero-shot TTS through speech prompting," in *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[7] Z. Ju, Y. Wang, K. Shen, X. Tan, D. Xin, D. Yang, E. Liu, Y. Leng, K. Song, S. Tang, Z. Wu, T. Qin, X. Li, W. Ye, S. Zhang, J. Bian, L. He, J. Li, and sheng zhao, "Naturalspeech 3: Zero-shot speech synthesis with factorized codec and diffusion models," in *Forty-first International Conference on Machine Learning*, 2024.

[8] 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 *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 6613–6617.

[9] S. Arik, J. Chen, K. Peng, W. Ping, and Y. Zhou, "Neural voice cloning with a few samples," in *Advances in Neural Information Processing Systems*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds., vol. 31. Curran Associates, Inc., 2018.

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

[11] C.-P. Hsieh, S. Ghosh, and B. Ginsburg, "Adapter-Based Extension of Multi-Speaker Text-To-Speech Model for New Speakers," in *Proc. INTERSPEECH*, 2023, pp. 3028–3032.

[12] W. Wang, Y. Song, and S. Jha, "Usat: A universal speaker-adaptive text-to-speech approach," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 32, pp. 2590–2604, 2024.

[13] N. Ruiz, Y. Li, V. Jampani, Y. Pritch, M. Rubinstein, and K. Aberman, "Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2023, pp. 22 500–22 510.

[14] S. Kim, H. Kim, and S. Yoon, "Guided-tts 2: A diffusion model for high-quality adaptive text-to-speech with untranscribed data," *arXiv preprint arXiv:2205.15370*, 2022.

[15] H. Kim, S. Kim, J. Yeom, and S. Yoon, "Unitspeech: Speaker-adaptive speech synthesis with untranscribed data," in *INTERSPEECH*, 2023, pp. 3038–3042.

[16] E. J. Hu, yelong shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "LoRA: Low-rank adaptation of large language models," in *International Conference on Learning Representations*, 2022.

[17] H. Kim, S. gil Lee, J. Yeom, C. H. Lee, S. Kim, and S. Yoon, "Voicetailor: Lightweight plug-in adapter for diffusion-based personalized text-to-speech," in *Interspeech*, 2024, pp. 4413–4417.

[18] X. L. Li and P. Liang, "Prefix-tuning: Optimizing continuous prompts for generation," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, C. Zong, F. Xia, W. Li, and R. Navigli, Eds. Online: Association for Computational Linguistics, Aug. 2021, pp. 4582–4597.

[19] R. Zhang, J. Han, C. Liu, A. Zhou, P. Lu, Y. Qiao, H. Li, and P. Gao, "LLaMA-adapter: Efficient fine-tuning of large language models with zero-initialized attention," in *The Twelfth International Conference on Learning Representations*, 2024.

[20] W. Kwon, Z. Li, S. Zhuang, Y. Sheng, L. Zheng, C. H. Yu, J. Gonzalez, H. Zhang, and I. Stoica, "Efficient memory management for large language model serving with pagedattention," in *Proceedings of the 29th Symposium on Operating Systems Principles*, ser. SOSP '23. New York, NY, USA: Association for Computing Machinery, 2023, p. 611–626.

[21] G.-I. Yu, J. S. Jeong, G.-W. Kim, S. Kim, and B.-G. Chun, "Orca: A distributed serving system for Transformer-Based generative models," in *16th USENIX Symposium on Operating Systems Design and Implementation (OSDI 22)*. Carlsbad, CA: USENIX Association, Jul. 2022, pp. 521–538.

[22] Y. Wen and S. Chaudhuri, "Batched low-rank adaptation of foundation models," in *The Twelfth International Conference on Learning Representations*, 2024.

[23] N. Kumari, B. Zhang, R. Zhang, E. Shechtman, and J.-Y. Zhu, "Multi-concept customization of text-to-image diffusion," in *CVPR*, 2023.

[24] Y. Gu, X. Wang, J. Z. Wu, Y. Shi, Y. Chen, Z. Fan, W. XIAO, R. Zhao, S. Chang, W. Wu, Y. Ge, Y. Shan, and M. Z. Shou, "Mix-of-show: Decentralized low-rank adaptation for multi-concept customization of diffusion models," in *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[25] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 6840–6851.

[26] 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 *Interspeech*, 2019, pp. 1526–1530.

[27] S. yang Liu, C.-Y. Wang, H. Yin, P. Molchanov, Y.-C. F. Wang, K.-T. Cheng, and M.-H. Chen, "DoRA: Weight-decomposed low-rank adaptation," in *Forty-first International Conference on Machine Learning*, 2024.

[28] D. J. Kopiczko, T. Blankevoort, and Y. M. Asano, "VeRA: Vector-based random matrix adaptation," in *The Twelfth International Conference on Learning Representations*, 2024.

[29] M. Nikdan, S. Tabesh, E. Crnčević, and D. Alistarh, "RoSA: Accurate parameter-efficient fine-tuning via robust adaptation," in *Forty-first International Conference on Machine Learning*, 2024.

[30] E. Casanova, K. Davis, E. Gölge, G. Göknar, I. Gulea, L. Hart, A. Aljafari, J. Meyer, R. Morais, S. Olayemi, and J. Weber, "Xtts: a massively multilingual zero-shot text-to-speech model," in *Interspeech*, 2024, pp. 4978–4982.

[31] Z. Du, Q. Chen, S. Zhang, K. Hu, H. Lu, Y. Yang, H. Hu, S. Zheng, Y. Gu, Z. Ma, Z. Gao, and Z. Yan, "Cosyvoice: A scalable multilingual zero-shot text-to-speech synthesizer based on supervised semantic tokens," *arXiv preprint arXiv:2407.05407*, 2024.

[32] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *International Conference on Learning Representations (ICLR)*, San Diega, CA, USA, 2015.

[33] S. gil Lee, W. Ping, B. Ginsburg, B. Catanzaro, and S. Yoon, "BigVGAN: A universal neural vocoder with large-scale training," in *The Eleventh International Conference on Learning Representations*, 2023.

[34] G. Louppe, "Resemblyzer," https://github.com/resemble-ai/Resemblyzer, 2019.

[35] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu, and R. Pang, "Conformer: Convolution-augmented Transformer for Speech Recognition," in *Proc. Interspeech*, 2020, pp. 5036–5040.

[36] O. Kuchaiev, J. Li, H. Nguyen, O. Hrinchuk, R. Leary, B. Ginsburg, S. Kriman, S. Beliaev, V. Lavrukhin, J. Cook *et al.*, "Nemo: a toolkit for building ai applications using neural modules," *arXiv preprint arXiv:1909.09577*, 2019.

[37] L. Zhang, L. Zhang, S. Shi, X. Chu, and B. Li, "Lora-fa: Memory-efficient low-rank adaptation for large language models fine-tuning," *arXiv preprint arXiv:2308.03303*, 2023.