VOICE FILTER: FEW-SHOT TEXT-TO-SPEECH SPEAKER ADAPTATION USING VOICE CONVERSION AS A POST-PROCESSING MODULE

Adam Gabryś⋆, *Goeric Huybrechts*⋆, *Manuel Sam Ribeiro*⋆, *Chung-Ming Chien*†, *Julian Roth*⋆, *Giulia Comini*⋆, *Roberto Barra-Chicote*⋆, *Bartek Perz*⋆, *Jaime Lorenzo-Trueba*⋆

⋆ Alexa AI † National Taiwan University (NTU)

ABSTRACT

State-of-the-art text-to-speech (TTS) systems require several hours of recorded speech data to generate high-quality synthetic speech. When using reduced amounts of training data, standard TTS models suffer from speech quality and intelligibility degradations, making training low-resource TTS systems problematic. In this paper, we propose a novel extremely low-resource TTS method called Voice Filter that uses as little as one minute of speech from a target speaker. It uses voice conversion (VC) as a post-processing module appended to a pre-existing high-quality TTS system and marks a conceptual shift in the existing TTS paradigm, framing the few-shot TTS problem as a VC task. Furthermore, we propose to use a duration-controllable TTS system to create a parallel speech corpus to facilitate the VC task. Results show that the Voice Filter outperforms state-of-the-art few-shot speech synthesis techniques in terms of objective and subjective metrics on one minute of speech on a diverse set of voices, while being competitive against a TTS model built on 30 times more data.[1]

***Index Terms*—** Text-To-Speech, Speaker Adaptation, Voice Conversion, Few-Shot Learning

1. INTRODUCTION

State-of-the-art text-to-speech (TTS) technologies are capable of generating high-quality synthetic speech on a variety of situations. In order to achieve very high quality, TTS typically requires several hours of studio-quality data, drawn from either a single or multiple speakers [1]. As such, reducing the amount of speech data to a few hours imposes limitations on the quality and intelligibility of those systems [2]. Because it is not always feasible to collect several hours of speech data, particularly when scaling TTS voices to a large number of new speakers, the problem of building low-resource TTS voices has been thoroughly explored [3, 4, 5, 6]. When the aim is building high-quality TTS systems in extremely low-resource scenarios, such as when only one minute of speech from the target speaker is available, we primarily aim to capture speaker identity and we must defer the modelling of phonetic and prosodic variability to supplementary speech data.

Common approaches to this problem therefore rely on speaker adaptation, whereby the parameters of a multi-speaker model are optimized on a few samples from the target speaker [7, 8, 9]. The adaptation process aims to modify only the speaker identity, which are the speech attributes defining the target speaker as an individual. To control speaker identity in few-shot speaker adaptation, there are techniques such as Vector-Quantized models [10, 11, 12], U-Net structures [13, 12], attention mechanisms [14] or a combination of loss functions [15, 16]. Alternatively, speaker identity can be controlled through representations that are inherited from external speaker verification systems or jointly trained with the primary model [17, 18, 16]. During adaptation, studies have proposed to optimize all model parameters [9, 19], selected components [20, 21, 22], or to focus instead on external speaker representations [9, 17]. An alternative approach is to address the adaptation problem via data augmentation. This can be done through conventional signal processing techniques [23], but there are more sophisticated methods that proposed producing high-quality synthetic data for the target speaker through a voice conversion (VC) model [6, 24]. The TTS system is then optimized from scratch on a mixture of natural and synthetic data.

There are, however, shortcomings associated with these methods. Wang et al [25] suggests that, with the speaker adaptation strategy, a single architecture becomes responsible for the modelling of linguistic content and speaker identity. As in this framework it is not entirely clear which model parameters are responsible for speaker identity, the impact of parameter adaptation can be diluted. Additionally, fine-tuning a complex architecture on a few samples can easily lead to model over-fitting, reducing overall quality and intelligibility. On the other hand, data augmentation-based approaches, still require at least 15 minutes of training data from the target speaker in order to optimise the TTS models. As such they aren't directly applicable to very low-resource scenarios.

In this paper, we address the problem of extremely low-resource TTS by using VC as a post-processing module which we refer to as "Voice Filter" on top of a high-quality single-speaker TTS model. This single-speaker TTS model is also used to generate a synthetic parallel corpus for the Voice Filter training. Our proposal presents the following novelties and advantages: (1) The overall process becomes modular, splitting it into speech content generation task followed by a speaker identity generation one. This improves efficiency, robustness and interpretability, as well as enabling task-dependent adaptation. (2) We leverage the strengths of parallel VC without assuming that a parallel corpus is available by synthetically generating frame-level matching speech pairs via a duration-controllable TTS model.

In summary, we split the problem of traditional few-shot TTS voice creation into two tasks: speech content and speaker identity generation. The split means we can lower the amount of speech required to train a synthetic voice for a particular speaker identity down to as little as one minute of speech by limiting the complexity of the problem. This translates in the produced synthetic speech quality being comparable to those of TTS models trained on 30 times more data.

[1] Audio samples will be made publicly available with a related blog post on www.amazon.science.

[†] Work carried out as an intern.

978-1-6654-0540-9/22/$31.00 ©2022 IEEE

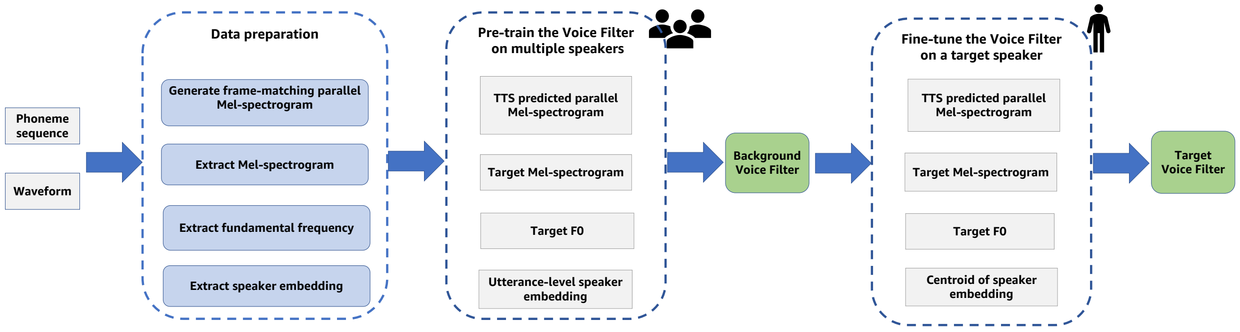
Phoneme sequence
Waveform
Data preparation
Generate frame-matching parallel Mel-spectrogram
Extract Mel-spectrogram
Extract fundamental frequency
Extract speaker embedding
Pre-train the Voice Filter on multiple speakers
TTS predicted parallel Mel-spectrogram
Target Mel-spectrogram
Target F0
Utterance-level speaker embedding
Background Voice Filter
Fine-tune the Voice Filter on a target speaker
TTS predicted parallel Mel-spectrogram
Target Mel-spectrogram
Target F0
Centroid of speaker embedding
Target Voice Filter

Fig. 1. Synthetic parallel data preparation and training flowchart for the proposed Voice Filter.

2. METHOD

Voice Filter approaches the problem of extremely low-resource TTS voice building by decoupling the speech content and the speaker identity generation into two separate tasks, with the Voice Filter itself focusing on the latter. This results in more modularization and more robust speaker identity generation compared to adapting a multi-speaker TTS model. Since the Voice Filter is in charge of speaker identity generation only, it operates at a lower abstraction level (Mel-spectrograms) than the overall TTS system (phonemes). We believe that this speech-to-speech task is easier than the text-to-speech one, especially in very low-resource settings.

The proposed VC module (Voice Filter) is located between a single-speaker duration-controllable TTS model [24] and a universal neural vocoder [26]. This enables us to generate Mel-spectrograms for any desired text with the TTS model, which is subsequently given the appropriate speaker identity by the Voice Filter and ultimately converted into a time-domain waveform by the vocoder.

2.1. Creation of synthetic parallel corpus

The proposed method begins by synthetically generating a parallel dataset to enable training of our Voice Filter (Figure 1, Data preparation). Working with a synthetic parallel corpus allows us to overcome two of the biggest limitations of parallel VC, while still leveraging its strengths: (1) It is no longer required to have a large parallel speech corpus between source and target speakers, which is hard and expensive to collect; and (2) It is no longer necessary to apply duration warping methods to align the parallel corpora across the multiple speakers. This is possible because the input to the Voice Filter is duration-controlled constructed synthetic speech and not recordings. However, this means that in order to generate the parallel corpus we require two different datasets: a single-speaker corpus for the TTS system and a multi-speaker corpus for the Voice Filter training.

Creating the synthetic parallel corpus consists of 3 steps:

1. Force-align all available data at the phone level, which we did using the pre-trained Kaldi ASpIRE TDNN system [27], including both the single-speaker and multi-speaker corpora.
2. Train a duration-controllable TTS system [24], making use of the single-speaker corpus.
3. Generate synthetic data matching the transcripts and phone-level durations of the multi-speaker corpus that was aligned in the preliminary step using the trained duration-controllable TTS system.

This three-step process results in a parallel corpus that is aligned at the frame-level between synthetic single-speaker and natural multi-speaker speech samples. It forms the training data for the proposed Voice Filter, which to the best of our knowledge, is a novel inclusion to the VC problem.

For the purpose of this paper and experimentation, the single-speaker corpus contains 10 hours of high-quality speech data read in a neutral speaking style by a male US English speaker. The multi-speaker corpus consists of 120 gender-balanced male and female US English speakers, with approximately 40 minutes of data per speaker with complete phonetic coverage.

2.2. Model architecture

The Voice Filter model (Figure 2) takes as inputs and outputs 80-bin Mel-spectrograms of equal-length and consists of a 6-stack of size-preserving 1D convolutions with 512 channels and a kernel size of 5 with batch-norm. This is followed by a uni-directional LSTM and a Dense layer with 1024 nodes.

We concatenate the target speaker embedding and log-f0 contour to the hidden representation of the third convolution layer. The speaker embedding is a 256-dimensional vector defined at the utterance level and broadcasted to the frame level. The speaker verification system used to extract the embeddings was trained on the multi-speaker corpus and optimized on a Generalized End-to-End Loss [28]. We observed that the log-f0 contour helps the model better absorb the prosodic differences between the input and target speaker. Effectively speaking, this means that the Voice Filter doesn't need to learn how to adjust prosody information between the source and the target speakers but rather focuses on speaker-defining information. To extract the log-f0 contour from the target speech recordings, we used the RAPT algorithm [29] of the Speech Processing Toolkit (SPTK[2]) with a threshold of 0 for voiced/unvoiced regions.

2.3. Model training and fine-tuning

Training a Voice Filter model that can generate speech from 1 minute of unseen speaker data is a two-step process: (1) background model training (Figure 1, Pre-train VF) and (2) fine-tuning to the minute of target unseen speaker data (Figure 1, Fine-tune VF).

[2] http://sp-tk.sourceforge.net

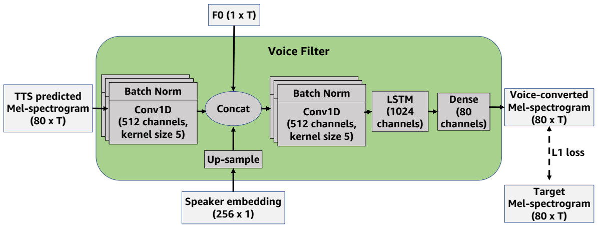
F0 (1 x T)
Voice Filter
TTS predicted Mel-spectrogram (80 x T)
Batch Norm
Conv1D (512 channels, kernel size 5)
Concat
Batch Norm
Conv1D (512 channels, kernel size 5)
LSTM (1024 channels)
Dense (80 channels)
Voice-converted Mel-spectrogram (80 x T)
Up-sample
L1 loss
Speaker embedding (256 x 1)
Target Mel-spectrogram (80 x T)

Fig. 2. Proposed Voice Filter architecture.

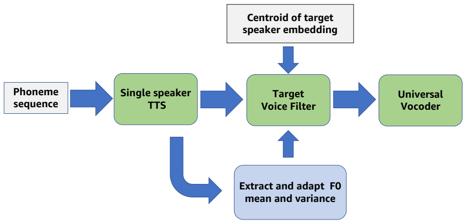
Centroid of target speaker embedding
Phoneme sequence
Single speaker TTS
Target Voice Filter
Universal Vocoder
Extract and adapt F0 mean and variance

Fig. 3. Proposed Voice Filter inference flowchart.

1. The background Voice Filter, is trained in a one-to-many fashion for 1 million steps on the entire synthetic parallel multi-speaker corpus generated in the previous step. This model is capable of converting to any of the speakers seen during training but isn't robust enough to generalize to unseen speakers without further processing.
2. We adapt the background model to the target Voice Filter by fine-tuning all the parameters of the background Voice Filter for 1000 steps on the target speaker's single minute of speech in a one-to-one fashion. We use the centroid of the target utterance-level speaker embeddings as we observed that, in our few-shot scenario, fine-tuning on a constant speaker embedding rather than on variable utterance-level embeddings resulted in more stable models. We have not tested the impact in quality on non-target speakers after fine-tuning, but we consider the resulting target Voice Filter to be speaker-dependent.

Both the background and target Voice Filter models are trained using the L1 spectral loss and the ADAM optimiser with default settings.

2.4. Model Inference

Inference on the complete model (Figure 3) requires us to run several models in succession:

1. Inference of the source Mel-spectrogram for the desired text and predicted durations with the single-speaker TTS model.
2. Estimation of f0 from the source Mel-spectrogram and re-normalization to the mean and variance of the target speaker.
3. Conversion of the source Mel-spectrogram to the target speaker by the fine-tuned Voice Filter.
4. Synthesis of the voice-converted Mel-spectrogram to time-domain waveform via the vocoder.

At this moment, we do not adapt speaking rate or phone durations to those of the target speaker as they are not simple to estimate on extremely low-resource scenarios and lead to significant artefacts.

3. EXPERIMENTAL SETUP

Models were assessed using both objective and perceptual metrics. For our evaluations we have used 4 male and 4 female speakers with 50 test utterances per speaker, resulting in 400 prompts overall.

Signal quality is objectively measured using the conditional Fréchet Speech Distance (cFSD) [30]. Specifically, a pre-trained XLSR-53 [31] wav2vec-2.0 [32] model is used to generate the activation distributions for recordings and synthesised samples. The distributions are then compared with a Fréchet Distance, which provides a measure of how close the generated speech is to an actual recording. To objectively estimate a speaker similarity metric, we used the mean cosine distance between speaker embeddings (CSED) of recordings and predicted samples.

MUltiple Stimuli with Hidden Reference and Anchor (MUSHRA) tests were used to perceptually assess naturalness, signal quality, speaker and speaking style similarity. Samples from the systems being evaluated were presented to participants side by side. They were asked to score them on a scale from 0 (the worst) to 100 (the best) in terms of the metric being evaluated. We used the crowd-sourcing platform ClickWorker to assess each test utterance by a panel of 25 listeners. Target speaker recordings were always included as a hidden upper-anchor system and we did not enforce requirements for at least one system to be rated 100. We provided the listeners with a reference sample for both the speaker and style similarity evaluations. The lower-anchor for the speaker similarity evaluation was the voice-converted samples of the furthest same gender speaker in the speaker embedding space. The lower-anchor for the style similarity evaluation was the un-filtered TTS system. Paired two-sided Student T-tests with Holm-Bonferroni correction were used to validate the statistical significance of the differences between two systems at a p-value threshold of 0.05.

4. RESULTS

4.1. Extreme-low resource speech synthesis performance

We compared our system with two multi-speaker state-of-the-art technologies that have shown high performance for low-resource speech synthesis: CopyCat (CC) [33] model with an additional f0 conditioning[3] and a duration-controllable multi-speaker TTS (MS-TTS) [24] model without the data augmentation component. Both models were trained on the same dataset and conditions described in section 2.3, including fine-tuning for 1000 steps on the 1 minute of target speaker data.

Objective metrics and perceptual MUSHRA evaluation scores are reported in tables 1 (rows 1-3) and 2, respectively. We observe a statistical preference in the MUSHRA evaluation for the proposed system in terms of all evaluated metrics. The objective metrics are aligned with these observations, indicating that our proposed method outperforms other speaker adaptation techniques when using the same amount of data.

[3] Preliminary results indicated that f0-conditioned CopyCat showed better signal quality stability and constituted a fairer comparison.

System	Speaker sim. (CSED)	Signal quality (cFSD)
VF	**0.192**	**0.197**
CC	0.198	0.249
MS-TTS	0.207	0.263
TTS-DA	0.205	0.224

Table 1. Average objective metrics for all evaluated systems. Best numbers are highlighted in bold. TTS-DA was trained on 30 minutes of speech instead of 1.

System	Sp. sim.	Style sim.	Nat.	Sig. Q.
Rec	79.43 ± 0.66	75.53 ± 0.73	78.72 ± 0.64	76.67 ± 0.75
VF	$\mathbf{67.96} \pm 0.86$	$\mathbf{64.4} \pm 0.82$	$\mathbf{53.09} \pm 0.81$	$\mathbf{56.28} \pm 0.84$
CC	66.57 ± 0.88	63.56 ± 0.84	52.08 ± 0.81	55.28 ± 0.85
MS-TTS	65.83 ± 0.91	62.10 ± 0.86	50.71 ± 0.81	54.24 ± 0.85
Lower-anchor	37.90 ± 1.00	37.96 ± 1.12	–	–

Table 2. Average MUSHRA results with confidence interval of 95%. Best scores with statistical difference between Voice filter (VF) and reference systems are highlighted in bold ($p < 0.05$).

4.2. Ablation study on data quantity

In order to understand the impact of the extremely low-resource scenario, we trained the Voice Filter using 1, 5 and 25 minutes of target speaker data during fine-tuning. Objective metrics and perceptual MUSHRA evaluation scores are reported in tables 3 and 4, respectively.

In the subjective evaluation, listeners did not perceive a statistically significant difference between the different data scenarios, although objective metrics still show slight improvements with bigger amounts of data. This hints that, while there may be room to improve the performance of the system, the nature of our Voice Filter doesn't benefit perceptually from richer data scenarios. With only 1 minute of target data, we are able to create high-quality samples.

# min	Speaker sim. (CSED)	Signal quality (cFSD)
1 min	0.192	0.197
5 min	**0.183**	0.189
25 min	0.185	**0.176**

Table 3. Average objective metrics for Voice Filter trained using varying amount of data. Best results are highlighted in bold.

# min	Sp. sim.	Style sim.	Nat.	Sig. Q.
Rec	73.77 ± 0.99	72.06 ± 0.95	78.69 ± 0.70	73.94 ± 0.91
1 min	51.90 ± 0.99	54.67 ± 0.98	55.10 ± 0.84	53.94 ± 0.95
5 min	51.93 ± 1.01	55.36 ± 0.97	55.06 ± 0.83	53.83 ± 0.97
25 min	52.00 ± 1.01	54.75 ± 0.99	55.21 ± 0.82	53.63 ± 0.95

Table 4. Average MUSHRA results with confidence interval of 95% for Voice Filter trained with varying amount of data. No statistically significant differences ($p < 0.05$) between systems.

4.3. Comparison against a competitive TTS

Finally, we investigate the quality of the generated speech when compared to a TTS system trained on a larger amount of target recordings. For that, we compared Voice Filter against a validated low-resource TTS technology that showed to be competitive when trained on 30 minutes of target speech (TTS-DA) [6]. It is worth noting that such a technology implicitly results in the TTS system estimating the phone duration for the target speaker, which is not the case for the proposed Voice Filter. Effectively this means we are comparing the proposed Voice Filter against a system trained on 30 times more target data that has been validated also to be competitive against TTS voices trained on 5+ hours of target recordings.

Objective metrics and perceptual MUSHRA evaluation scores are reported in tables 1 (rows 1 & 4) and 5, respectively. While we observe a 4% relative degradation in speaker similarity, the MUSHRA evaluations indicate no statistical difference between the systems in terms of signal quality, naturalness, and style similarity. On the other hand, in terms of objective metrics, we observe that speaker similarity and signal quality are better for our proposed method. Overall, results show that our model is on par with TTS-DA with a slight human-perceived degradation in terms of speaker similarity, and this despite the much smaller target training dataset being used.

System	Sp. sim.	Style sim.	Nat.	Sig. Q.
Rec	83.23 ± 0.66	79.82 ± 0.80	77.05 ± 0.56	76.75 ± 0.52
VF	70.38 ± 0.93	69.53 ± 0.98	55.24 ± 0.80	55.60 ± 0.72
TTS-DA	$\mathbf{73.67} \pm 0.85$	69.97 ± 1.01	55.21 ± 0.81	55.83 ± 0.71
Lower-anchor	37.74 ± 1.10	39.51 ± 1.46	–	–

Table 5. Average MUSHRA results. Best scores with statistical difference between Voice filter (VF) and reference systems are highlighted in bold ($p < 0.05$)

5. CONCLUSIONS

In this work, we proposed a novel extremely low-resource TTS method called Voice Filter that can produce high-quality speech when using only 1 minute of speech.

Voice Filter splits the TTS process in a speech content and speaker identity generation task. The speaker identity is generated via a fine-tuned one-to-many VC module, which makes it easily scalable to new speakers even on extremely low-resource settings. The speech content generation module is a duration-controllable single-speaker TTS system that has the added benefit of enabling us to generate a synthetic parallel corpus. This enables Voice Filter to work in a parallel frame-level condition, which has a higher quality ceiling and lower modelling complexity. Evaluations show that our Voice Filter outperforms other few-shot speech synthesis techniques in terms of objective and subjective metrics on the 1-minute data scenario, with quality comparable to a SOTA system trained on 30 times more data.

In conclusion, we consider the Voice Filter paradigm to be a first step towards building extremely low-resource TTS as a post-processing VC plug-in. Moreover, we believe that the generation of synthetic parallel duration-controllable data will enable further scenarios in speech technologies that were limited previously because of data availability.

6. REFERENCES

[1] J. Latorre, J. Lachowicz, J. Lorenzo-Trueba, et al., “Effect of data reduction on sequence-to-sequence neural tts,” in *Proc ICASSP*. IEEE, 2019, pp. 7075–7079.

[2] Y.-A. Chung, Y. Wang, W.-N. Hsu, Y. Zhang, and R. Skerry-Ryan, “Semi-supervised training for improving data efficiency in end-to-end speech synthesis,” in *Proc. ICASSP)*. IEEE, 2019, pp. 6940–6944.

[3] Y.-J. Chen, T. Tu, C. chieh Yeh, and H.-Y. Lee, “End-to-End Text-to-Speech for Low-Resource Languages by Cross-Lingual Transfer Learning,” in *Proc. Interspeech 2019*, 2019, pp. 2075–2079.

[4] H. Zhang and Y. Lin, “Unsupervised Learning for Sequence-to-Sequence Text-to-Speech for Low-Resource Languages,” in *Proc. Interspeech 2020*, 2020, pp. 3161–3165.

[5] J. Xu, X. Tan, Y. Ren, et al., “Lrspeech: Extremely low-resource speech synthesis and recognition,” in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020, pp. 2802–2812.

[6] G. Huybrechts, T. Merritt, G. Comini, et al., “Low-resource expressive text-to-speech using data augmentation,” in *Proc. ICASSP*. IEEE, 2021, pp. 6593–6597.

[7] Q. Xie, X. Tian, G. Liu, et al., “The multi-speaker multi-style voice cloning challenge 2021,” in *Proc. ICASSP*, 2021, pp. 8613–8617.

[8] Y. Taigman, L. Wolf, A. Polyak, and E. Nachmani, “Voiceloop: Voice fitting and synthesis via a phonological loop,” in *International Conference on Learning Representations*, 2018.

[9] Y. Chen, Y. Assael, B. Shillingford, et al., “Sample efficient adaptive text-to-speech,” in *International Conference on Learning Representations*, 2019.

[10] A. van den Oord, O. Vinyals, and k. kavukcuoglu, “Neural discrete representation learning,” in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, et al., Eds. 2017, vol. 30, Curran Associates, Inc.

[11] A. Razavi, A. van den Oord, and O. Vinyals, “Generating diverse high-fidelity images with vq-vae-2,” in *Advances in neural information processing systems*, 2019, pp. 14866–14876.

[12] D.-Y. Wu, Y.-H. Chen, and H. yi Lee, “VQVC+: One-Shot Voice Conversion by Vector Quantization and U-Net Architecture,” in *Proc. Interspeech 2020*, 2020, pp. 4691–4695.

[13] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

[14] 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*, 2020, pp. 2007–2011.

[15] T. Wang, J. Tao, R. Fu, et al., “Bi-level speaker supervision for one-shot speech synthesis.,” in *Proc. Interspeech*, 2020, pp. 3989–3993.

[16] Z. Cai, C. Zhang, and M. Li, “From Speaker Verification to Multispeaker Speech Synthesis, Deep Transfer with Feedback Constraint,” in *Proc. Interspeech 2020*, 2020, pp. 3974–3978.

[17] 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*, 2018, vol. 31.

[18] Y. Jia, Y. Zhang, R. J. Weiss, et al., “Transfer learning from speaker verification to multispeaker text-to-speech synthesis,” in *Advances in Neural Information Processing Systems 31*, 2018, pp. 4485–4495.

[19] Z. Kons, S. Shechtman, A. Sorin, C. Rabinovitz, and R. Hoory, “High Quality, Lightweight and Adaptable TTS Using LPC-Net,” in *Proc. Interspeech 2019*, 2019, pp. 176–180.

[20] H. B. Moss, V. Aggarwal, N. Prateek, J. González, and R. Barra-Chicote, “Boffin tts: Few-shot speaker adaptation by bayesian optimization,” in *Proc. ICASSP*. IEEE, 2020, pp. 7639–7643.

[21] 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.

[22] M. Chen, X. Tan, B. Li, et al., “Adaspeech: Adaptive text to speech for custom voice,” *International Conference on Learning Representations (ICLR)*, 2021.

[23] B. Lorincz, A. Stan, and M. Giurgiu, “Speaker verification-derived loss and data augmentation for dnn-based multispeaker speech synthesis,” in *29th European Signal Processing Conference, EUSIPCO 2021, Dublin, Ireland, August 23-27, 2021*. 2021, pp. 26–30, IEEE.

[24] R. Shah, K. Pokora, A. Ezzerg, et al., “Non-Autoregressive TTS with Explicit Duration Modelling for Low-Resource Highly Expressive Speech,” in *Proc. 11th ISCA Speech Synthesis Workshop (SSW 11)*, 2021, pp. 96–101.

[25] T. Wang, J. Tao, R. Fu, et al., “Spoken content and voice factorization for few-shot speaker adaptation.,” in *Proc. Interspeech*, 2020, pp. 796–800.

[26] A. Oord, Y. Li, I. Babuschkin, et al., “Parallel wavenet: Fast high-fidelity speech synthesis,” in *International conference on machine learning*. PMLR, 2018, pp. 3918–3926.

[27] V. Peddinti, G. Chen, V. Manohar, et al., “Jhu aspire system: Robust lvcsr with tdnns, ivector adaptation and rnn-lms,” in *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*. IEEE, 2015, pp. 539–546.

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

[29] D. Talkin and W. B. Kleijn, “A robust algorithm for pitch tracking (RAPT),” *Speech coding and synthesis*, vol. 495, pp. 518, 1995.

[30] M. Bińkowski, J. Donahue, S. Dieleman, et al., “High fidelity speech synthesis with adversarial networks,” in *International Conference on Learning Representations*, 2020.

[31] M. Ott, S. Edunov, A. Baevski, et al., “fairseq: A fast, extensible toolkit for sequence modeling,” in *Proceedings of NAACL-HLT 2019: Demonstrations*, 2019.

[32] 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*, 2020, vol. 33, pp. 12449–12460.

[33] S. Karlapati, A. Moinet, A. Joly, et al., “Copycat: Many-to-many fine-grained prosody transfer for neural text-to-speech,” in *Proc. Interspeech*, 2020.