



Adversarial and Sequential Training for Cross-lingual Prosody Transfer TTS

Min-Kyung Kim and Joon-Hyuk Chang

Department of Electronic Engineering
Hanyang University, Seoul, Republic of Korea

kmk9266@gmail.com, jchang@hanyang.ac.kr

This study presents a method for improving the performance of the text-to-speech (TTS) model by using three global speech-style representations: language, speaker, and prosody. Synthesizing different languages and prosody in the speaker's voice regardless of their own language and prosody is possible. To construct the embedding of each representation conditioned in the TTS model such that it is independent of the other representations, we propose an adversarial training method for the general architecture of TTS models. Furthermore, we introduce a sequential training method that includes rehearsal-based continual learning to train complex and small amounts of data without forgetting previously learned information. The experimental results show that the proposed method can generate good-quality speech and yield high similarity for speakers and prosody, even for representations that the speaker in the dataset does not contain.

Index Terms: text-to-speech, cross-lingual, prosody, adversarial training, continual learning

1. Introduction

Recent end-to-end text-to-speech (TTS) systems [1-2] have achieved good results in terms of generating human-like speech. TTS models have exhibited the ability to transfer style representations such as speakers [3-4], language [5], and prosody [6]. By extending such models, many researchers have investigated multispeaker multilingual [7-8] and multispeaker prosody [9] models. Because these representations are prone to entanglement, these are learned separately by models. In [7], the speaker's identity was preserved during language conversion by using the L1 consistency loss term. In [10], the adversarial loss was applied to extract language-independent speaker embedding and speaker-independent language embedding.

In previous works, transfer learning methods have been employed to adapt to new speakers or languages by using a small amount of data [4], [11-12]. One of the methods for speaker or language adaptation is fine-tuning the pre-trained model. This approach can reduce training costs and improve speaker-speech quality by using a limited amount of data. However, when the model is fine-tuned to adapt to new speakers, previously learned information can disappear; this is referred to as the catastrophic forgetting problem. Continual learning approaches have been utilized to solve this problem [13]; these are categorized into three types: regularization-based [14], data-rehearsal-based [15-16], and parameter-isolation-based [17]. In [18], continuous speaker adaptation based on a data-rehearsal-based continual learning method was proposed using only one speaker sample per batch. In [19], continual learning of multilingual TTS synthesis was proposed to add new languages by employing random sampling and weighted sampling based on data replay. These methods can continually add new information with small amounts of data to a well-trained pre-trained TTS model

and successfully preserve the previous information.

In this paper, we introduce novel methods for synthesizing speech, including three global speech-style representations: language, speaker, and prosody. Synthesizing different languages and prosody while maintaining speaker identity regardless of their own language and prosody is possible. Our contributions are summarized as follows:

- We design a general architecture for TTS, including language, speaker, and prosody representations.
- We propose an efficient method for training each representation disentangled with other representations using adversarial training.
- We present a novel sequential training of the rehearsal-based continual learning technique, which trains representations one-at-a-time even when the amount of data is unbalanced and small, and the representation is complex.

2. Methodology

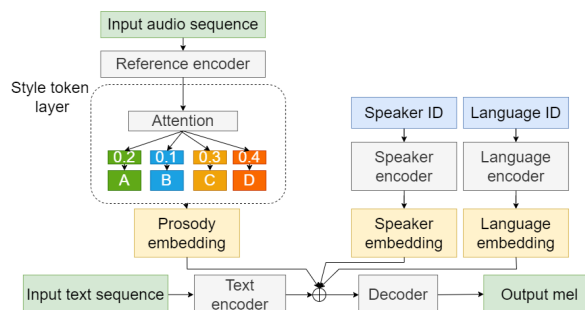


Figure 1: *Baseline model architecture for synthesizing speech with three global representations.*

2.1. Baseline

Fig. 1 presents the baseline model architecture for synthesizing speech with three global representations: language, speaker, and prosody. The model comprises two parts: a speech synthesizer and representation embeddings. The speech synthesizer is based on Tacotron 2 [2], which is an attention-based sequence-to-sequence model that predicts a mel spectrogram from an input text sequence. The representation embeddings conditioned on the speech synthesizer indicate embeddings that express the speech style, including language, speaker, and prosody. The language, speaker, and prosody embeddings represent language, speaker, and prosody information, respectively. During training, the speaker and language embeddings are obtained as

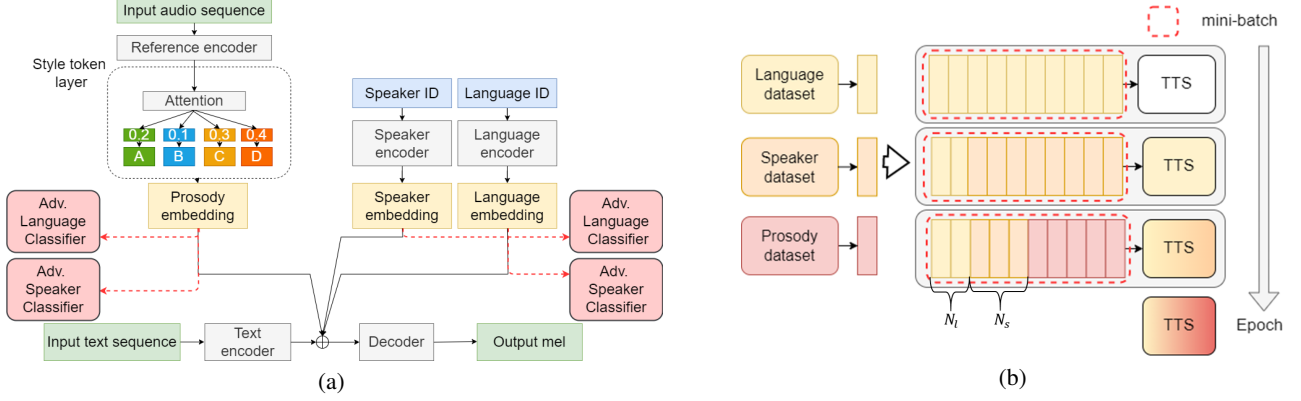


Figure 2: (a) Model architecture with adversarial training. (b) Block diagram of sequential training in the order of language, speaker, and prosody. One sample per representation’s label from datasets of previously learned representations is randomly included in a mini-batch, and the rest of the mini-batch consists of the data used for learning the representations.

the outputs of each encoder, as in [20], and the prosody embedding is the output of the global-style token layer proposed in GST-Tacotron [6].

The objective function to train the baseline can be formulated as a mel spectrogram L1 loss on a mini-batch as follows:

$$L_{mel} = \frac{1}{B} \sum_i^B |Y_i - \hat{Y}_i|, \quad (1)$$

where B , \hat{Y} , Y and are the mini-batch size, ground-truth mel spectrogram, and predicted mel spectrogram, respectively.

2.2. Adversarial training

For models that can make speakers pronounce different languages and prosody, embeddings corresponding to each speech representation must be extracted from the training data. However, representations in audio data are prone to entanglement, and the embeddings can represent or contain other representations, as explained in the following example. Language embeddings can be expressed as speaker embeddings when few speakers are present for one language. Moreover, if a few speakers have various forms of prosody, prosody embeddings may be described as the speaker’s embedding. We propose domain adversarial training [21] to train each representation separately, thereby avoiding the capture of other information, as illustrated in Fig. 2(a). We first denote the speaker classifier for language embeddings, language classifier for speaker embeddings, speaker classifier for prosody embeddings, and language classifier for prosody embeddings as G_{spk} , G_{lan} , G_{spk_p} , and G_{lan_p} , respectively.

The domain adversarial training objective of language embedding l to form an independent speaker embedding can be expressed as follows:

$$L_i^{ADV_{spk}} = - \sum_j^{N_s} \hat{s}_i^{(j)} \log (G_{spk}^{(j)} (\Delta l_i)), \quad (2)$$

where N_s and \hat{s} are the number of speakers and ground truths of the speaker label, respectively, and Δ is the gradient reversal operation. Similarly, the domain adversarial training objective of speaker embedding s to build an independent language rep-

resentation is defined as follows:

$$L_i^{ADV_{lan}} = - \sum_z^{N_l} \hat{l}_i^{(z)} \log (G_{lan}^{(z)} (\Delta s_i)), \quad (3)$$

where N_l denotes the number of languages used. Here, prosody is treated as the rest of the representations because its variability is unexplained and not easy to label among the three global representations. The domain adversarial training objective of prosody embedding p is formulated as follows:

$$L_i^{ADV_{pro}} = - \sum_j^{N_s} \hat{s}_i^{(j)} \log (G_{spk_p}^{(j)} (\Delta p_i)) - \sum_z^{N_l} \hat{l}_i^{(z)} \log (G_{lan_p}^{(z)} (\Delta p_i)). \quad (4)$$

To this end, the domain adversarial training objective is written as follows:

$$L_{adv} = \frac{1}{B} \sum_i^B [\lambda_1 L_i^{ADV_{spk}} + \lambda_2 L_i^{ADV_{lan}} + \lambda_3 L_i^{ADV_{pro}}], \quad (5)$$

where λ_1 , λ_2 , and λ_3 represent hyperparameters. Overall, the total loss for cross-lingual prosody transfer is formulated as follows:

$$L_{total} = L_{mel} + L_{adv}. \quad (6)$$

2.3. Sequential training

Obtaining data from bilingual expressive speakers is both expensive and challenging. Moreover, balancing the data for each language, speaker, and prosody representation is challenging. If the data are unbalanced, the model is inevitably trained to focus mainly on expression data, and learning low-resource data becomes challenging. To solve this, we introduce a method for sequentially training complex information by exploiting continual learning [14-19], which is a method used to upgrade one model gradually and handle multiple tasks. We chose to learn three complex representations by sequentially upgrading from

Table 2: Results of MOS: (a) and (b) audios used for evaluation were synthesized in intra-lingual and cross-lingual speakers’ prosody, respectively.

(a)												
	Naturalness				Speaker similarity				Prosody similarity			
	Intra-lingual		Cross-lingual		Intra-lingual		Cross-lingual		Intra-lingual		Cross-lingual	
	EN	KR	EN	KR	EN	KR	EN	KR	EN	KR	EN	KR
Baseline	3.19	3.22	2.75	2.61	3.44	3.67	3.31	2.98	3.21	3.41	2.95	3.12
ADV	3.12	3.23	2.78	2.64	3.52	3.72	3.35	3.21	3.28	3.45	2.89	3.41
ADV-SEQ1	3.07	3.19	2.69	2.62	3.61	3.70	3.40	3.15	3.19	3.40	2.88	3.20
ADV-SEQ2	3.06	2.91	2.65	2.60	3.46	3.62	3.20	3.02	3.44	3.42	3.19	3.42

(b)												
	Naturalness				Speaker similarity				Prosody similarity			
	Intra-lingual		Cross-lingual		Intra-lingual		Cross-lingual		Intra-lingual		Cross-lingual	
	EN	KR	EN	KR	EN	KR	EN	KR	EN	KR	EN	KR
Baseline	2.86	3.12	2.65	2.61	3.25	3.41	3.25	2.94	2.91	2.71	2.65	3.14
ADV	2.91	3.10	2.66	2.62	3.41	3.52	3.32	2.92	3.21	3.12	2.78	3.27
ADV-SEQ1	2.72	2.91	2.65	2.59	3.48	3.43	3.34	3.13	3.12	2.79	2.64	2.85
ADV-SEQ2	2.83	2.90	2.61	2.51	3.20	3.31	3.19	2.71	3.19	3.15	3.02	3.30

a cross-lingual model to a cross-lingual multi-speaker prosody model. This model can create the representation we choose to learn at the corresponding stage and reduce training costs once we have a pre-trained model.

Motivated by [18], in which a method for adapting to a new speaker without forgetting previously learned speakers was proposed, we randomly included one sample of the previous stage data in the mini-batch to sequentially learn a representation of the corresponding stage without forgetting previously learned representations. Fig. 2(b) shows an example of the proposed sequential learning in the order of the language, speaker, and prosody. In the first step, learning is performed by using datasets of a single speaker corresponding to each language. In the next step, training is conducted using the multi-speaker data and part of the data used in the previous step. When the prosody data are trained in three steps, all datasets, including the language, speaker, and prosody, are used together. However, the data learned in the previous step randomly consisted of only one sample per speaker label in a mini-batch.

3. Experiments

3.1. Datasets and experimental setup

Table 1: Datasets

Language	Dataset	Speaker	Hours
English	LJ	1	11
	Libri-TTS	24	4
	2013 Blizzard Challenge dataset	1	10
Korean	NEU1	1	10
	NEU2	6	20
	EXP	1	12

The details of the dataset used for the training are listed in Table 1. We trained the models using the English dataset from LJ [22], Libri-TTS [23], and the 2013 Blizzard Challenge dataset [24]. In addition, we used proprietary Korean datasets denoted by NEU, NEU2, and EXP. Blizzard and EXP are datasets of an-

imation and expressive style utterances, whereas LJ, Libri-TTS, NEU1, and NEU2 are public emotional utterances. When sequential training was performed, the datasets were divided into the following to learn the representations at each stage: 1) language: LJ and NEU1, 2) speaker: Libri-TTS and Korean NEU2, and 3) prosody: Blizzard and EXP datasets. Only 2–12 s of utterance audio clips were included. All audio clips were down-sampled to 16 kHz and converted into an 80-dimensional mel spectrogram.

All adversarial classifiers were 2-layer fully-connected layers with 256 hidden units. The model was jointly trained with a batch size of 256. We used the Adam optimizer with an initial learning rate of 10^{-3} . Hyperparameters λ_1 , λ_2 , and λ_3 were increased from 0 to 0.05. We used the Griffin-Lim [25] algorithm as a vocoder that transforms the mel spectrogram into a waveform because we considered that evaluating the differences between the baseline and proposed models is sufficient.

To assess the performance, we trained the following six methods:

- Baseline: The general architecture and process of the baseline are illustrated in Fig. 1.
- ADV: Training baseline with adversarial training
- SEQ1: Training baseline with sequential training in the order of language, prosody, and speaker
- SEQ2: Training baseline with sequential training in the order of language, speaker, and prosody
- ADV-SEQ1: Training baseline with adversarial training and sequential training in the order of language, prosody, and speaker
- ADV-SEQ2: Training baseline with adversarial training and sequential training in the order of language, speaker, and prosody

When the speaker and language datasets were used as previously trained data in the sequential training method, one sample corresponding to each label was randomly imported and used in the mini-batch. Similarly, in the case of prosody data, it was randomly consisted of 1/4 of the mini-batch.

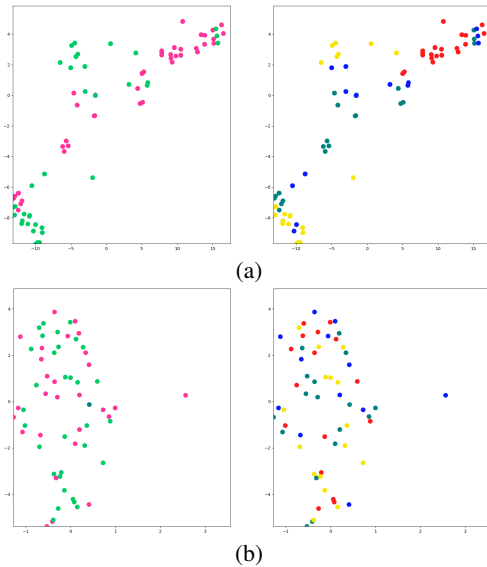


Figure 3: *t*-SNE visualization of embeddings. Embedding space of (a) baseline model and (b) ADV model. Starting from the left, the images show prosody embedding according to the language ID and the speaker ID. Each color represents different language and speaker ID.

3.2. Experimental results

3.2.1. Mean opinion score (MOS)

We determined the crowd-sourced MOS speech naturalness, speaker similarity, and prosody similarity to evaluate the synthesized speech, as shown in Table 2.

Naturalness: We evaluated the naturalness of the synthesized speech sentences. The overall score achieved in this study was lower than that obtained in previous studies because we used the Griffin-Lim algorithm as a vocoder. In the intra-lingual case, the baseline model showed scores similar to those of our methods. However, in most cross-lingual cases, our methods showed high performance.

Speaker similarity: We assessed the speaker similarity of the synthesized speech sentences. ADV and ADV-SEQ1 showed higher speaker similarity than the baseline in most cases. ADV-SEQ1 exhibited a somewhat higher performance when synthesized in other representations that the speaker did not have in the training data. This improvement may be attributed to the fact that SEQ1 was performed by using the speaker dataset in the previous step.

Prosody similarity: We evaluated the prosody similarity of the synthesized speech sentences. Similar to speaker similarity, ADV-SEQ2, which was the last to learn the prosody data, showed the highest performance.

Overall, the test results showed that learning by adversarial training increased naturalness and similarity. ADV-SEQ1 and ADV-SEQ2, which involved sequential learning, exhibited higher performance in terms of similarity when the representation that was learned at the end and the speaker in the dataset did not contain was transferred.

3.2.2. Adversarial training

We investigated the latent space for the language, speaker, and prosody embeddings generated by our model. We visualized the language, speaker, and prosody embeddings with the base-

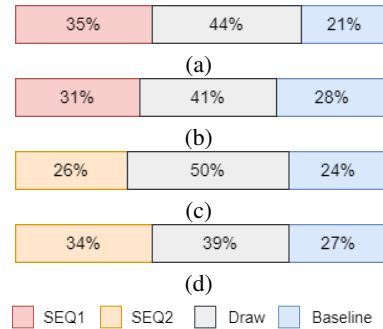


Figure 4: *AB preference results:* (a), (b) results of speaker similarity when there the amount of speaker data is small and (c), (d) results of prosody similarity with low-resource prosody data. (a), (c) were evaluated in the inter-lingual case, whereas (b), (d) were assessed in the cross-lingual case.

line and ADV by applying the t-distributed stochastic neighbor embedding (t-SNE) algorithm [26]. As shown in Fig. 3(a), the prosody embeddings for the language and speaker labels are easily entangled in case of the baseline. This implies that the prosody embeddings are related to other representations. Once our method is used, prosody embeddings are mixed to form representations that are unrelated to the speaker and language spaces, as shown in Fig. 3(b). This demonstrates that adversarial training effectively constructs representations and disentangles them from other representations.

3.2.3. Sequential training

We conducted an AB preference test to determine whether sequential training is practical when fewer data points correspond to each representation, namely, speaker or prosody. For the experiments depicted in Fig. 4(a), (b), 0.2 h of audio per speaker were included in the speaker dataset. Moreover, 1 h of audio per speaker was included in the prosody dataset, as shown in Fig. 4(c), (d). Our proposed method performed better than the baseline model in both the cross-lingual and intra-lingual cases. This indicates that sequential training with a few data points corresponding to the representation can capture the representation that we attempt to transfer to other representations.

4. Conclusions

We proposed a cross-lingual prosody TTS synthesis method that can disentangle three global representations of speech, namely, language, speaker, and prosody, and synthesize speech by combining these representations. Experiments showed that adversarial training can effectively extract the disentangled language, speaker, and prosody embeddings. Moreover, sequential training is also practical when the amount of data corresponding to the representations is small and unbalanced. In future work, we plan to investigate methods for adding new speakers and languages based on pretrained models without forgetting the previously learned representations.

5. Acknowledgements

This work was supported by Institute of Information communications Technology Planning Evaluation (IITP) grant funded by the Korea government(MSIT) (No. 2017-0-00474, Intelligent Signal Processing for AI Speaker Voice Guardian)

6. References

- [1] Y. Ren, Y. Ruan, X. Tan, T. Qin, S. Zhao, Z. Zhao, and T.-Y. Liu, "Fastspeech: Fast, robust and controllable text to speech," in *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, 2019, pp. 3165–3174.
- [2] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerry-Ryan, R. A. Saurous, Y. Agiomvrgiannakis, and Y. Wu, "Natural TTS synthesis by conditioning wavenet on mel spectrogram predictions," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 4779–4783.
- [3] A. Gibiansky, S. Arik, G. Diamos, J. Miller, K. Peng, W. Ping, J. Raiman, and Y. Zhou, "Deep voice 2: Multi-speaker neural text-to-speech," in *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, 2017, pp. 2962–2970.
- [4] Y. Jia, Y. Zhang, R. Weiss, Q. Wang, J. Shen, F. Ren, P. Nguyen, R. Pang, I. Lopez Moreno, and Y. Wu, "Transfer learning from speaker verification to multispeaker text-to-speech synthesis," in *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, 2018, pp. 4485–4495.
- [5] H. Ming, Y. Lu, Z. Zhang, and M. Dong, "A light-weight method of building an LSTM-RNN based bilingual TTS system," in *Proc. International Conference on Asian Language Processing (IALP)*, 2017, pp. 201–205.
- [6] 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. International Conference on Machine Learning (ICML)*, 2018, pp. 5180–5189.
- [7] E. Nachmani and L. Wolf, "Unsupervised polyglot text-to-speech," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 7055–7059.
- [8] Y. Zhang, R. J. Weiss, H. Zen, Y. Wu, Z. Chen, R. Skerry-Ryan, Y. Jia, A. Rosenberg, and B. Ramabhadran, "Learning to speak fluently in a foreign language: Multilingual speech synthesis and cross-language voice cloning," in *Proc. INTERSPEECH*, 2019, p. 2080–2084.
- [9] 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. International Conference on Machine Learning (ICML)*, 2018, pp. 4693–4702.
- [10] D. Xin, T. Komatsu, S. Takamichi, and H. Saruwatari, "Disentangled speaker and language representations using mutual information minimization and domain adaptation for cross-lingual TTS," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 6608–6612.
- [11] Y. Lee, S. Shon, and T. Kim, "Learning pronunciation from a foreign language in speech synthesis networks," *arXiv:1811.09364*, 2018.
- [12] Y. Chen, Y. Assael, B. Shillingford, D. Budden, S. Reed, H. Zen, Q. Wang, L. C. Cobo, A. Trask, B. Laurie, C. Gulcehre, A. V. D. Oord, O. Vinyals, and N. D. Freitas, "Sample efficient adaptive text-to-speech," in *Proc. International Conference on Learning Representations (ICLR)*, 2019.
- [13] R. M. French, "Catastrophic forgetting in connectionist networks," *Trends in Cognitive Sciences*, vol. 3, no. 4, pp. 128–135, 1999.
- [14] Z. Li and D. Hoiem, "Learning without forgetting," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 12, pp. 2935–2947, 2017.
- [15] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert, "iCaRL: Incremental classifier and representation learning," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 2001–2010.
- [16] D. Lopez-Paz and M. Ranzato, "Gradient episodic memory for continual learning," in *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, 2017, pp. 6470–6479.
- [17] J. Xu and Z. Zhu, "Reinforced continual learning," in *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, 2018, pp. 907–916.
- [18] H. Hemati and D. Borth, "Continual speaker adaptation for text-to-speech synthesis," *arXiv:2103.14512*, 2021.
- [19] M. Yang, S. Ding, T. Chen, T. Wang, and Z. Wang, "Towards life-long learning of multilingual text-to-speech synthesis," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 8022–8026, 2022.
- [20] J. Yang and L. He, "Towards universal text-to-speech," in *Proc. INTERSPEECH*, 2020, pp. 3171–3175.
- [21] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, "Domain-adversarial training of neural networks," *Journal of Machine Learning Research*, vol. 17, no. 1, pp. 2096–2030, 2016.
- [22] K. Ito and L. Johnson, "The LJ speech dataset," <https://keithito.com/LJ-Speech-Dataset/>, 2017.
- [23] 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*, pp. 1526–1530, 2019.
- [24] S. King and V. Karaiskos, "The blizzard challenge 2013," 2014.
- [25] D. Griffin and J. Lim, "Signal estimation from modified short-time fourier transform," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1984, pp. 236–243.
- [26] L. Van der Maaten and G. Hinton, "Visualizing data using t-SNE," *Journal of Machine Learning Research*, vol. 9, no. 11, pp. 2579–2605, 2008.