

Cotatron: Transcription-Guided Speech Encoder for Any-to-Many Voice Conversion without Parallel Data

Seung-won Park^{1,2}, Doo-young Kim^{1,2}, Myun-chul Joe²

¹Seoul National University, South Korea

²MINDs Lab Inc., South Korea

{swpark, dykim, mcjoe}@mindslab.ai

Abstract

We propose *Cotatron*, a transcription-guided speech encoder for speaker-independent linguistic representation. Cotatron is based on the multispeaker TTS architecture and can be trained with conventional TTS datasets. We train a voice conversion system to reconstruct speech with Cotatron features, which is similar to the previous methods based on Phonetic Posteriorgram (PPG). By training and evaluating our system with 108 speakers from the VCTK dataset, we outperform the previous method in terms of both naturalness and speaker similarity. Our system can also convert speech from speakers that are unseen during training, and utilize ASR to automate the transcription with minimal reduction of the performance. Audio samples are available at <https://mindslab-ai.github.io/cotatron>, and the code with a pre-trained model will be made available soon.

Index Terms: voice conversion, speech synthesis, speech representation, disentangled representation.

1. Introduction

Recent advances in voice conversion (VC) have shown potential for a wide variety of applications, such as enhancement of impaired speech or entertainment purposes. To switch the source speech’s speaker identity to that of the target speaker, the system should be able to encode speaker-independent (e.g., linguistic) features from given speech, and then pair them with a speaker representation to reconstruct the speech. Phonetic Posteriorgram (PPG) [1], a speaker-independent feature extracted with speaker-independent ASR, had been widely used for non-parallel voice conversion [2, 3, 4, 5]. However, PPG-based methods usually required additional acoustic features from audio analysis, which may indicate that the PPG itself is insufficient to encode rich linguistic features of human speech.

One way to encode speaker-independent features without discarding essential factors of the speech is to train a speech encoder with some restrictions. For example, Qian *et al.* [6] showed that an autoencoder with a carefully tuned bottleneck can effectively encode speaker-independent features without losing content information. Other prior works on restricting the speech encoders include: propagating reversed gradient from the speaker classifier [7], applying instance normalization [8], quantizing the representation [9, 10, 11], and training a conditioned flow-based generative model [12]. However, these methods required the model to discover linguistic representations by itself, although transcriptions were available within the dataset.

Initial attempts on incorporating text supervision to voice conversion system trained an auxiliary ASR decoder [13, 14, 15], or shared the model weights with TTS [16]. Unfortunately, these methods only dealt with a limited number of speakers or required huge amounts of data for each speaker; making their effectiveness on the real-world applications questionable.

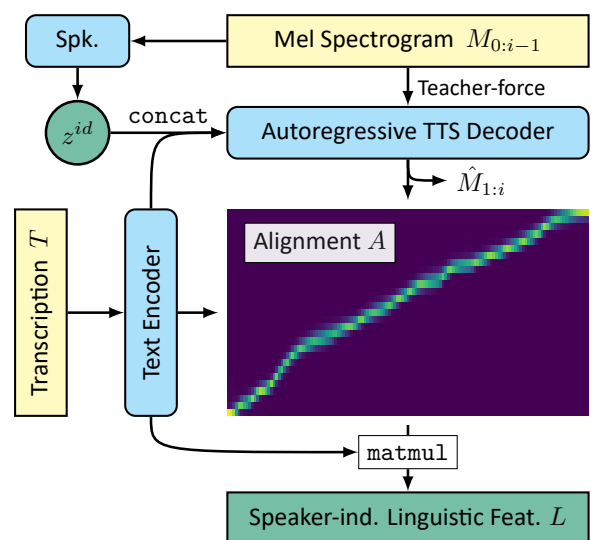


Figure 1: *Cotatron* architecture. The alignment between mel spectrogram and its transcription is obtained via a pre-trained multispeaker TTS (Tacotron2) and then combined with text encoding to extract speaker-independent linguistic features. Spk. denotes the speaker encoder.

In this paper, we propose *Cotatron*, a transcription-guided speech encoder based on a pre-trained multispeaker TTS model [17, 18]. Cotatron encodes an arbitrary speaker’s speech into speaker-independent linguistic features, which are fed to a decoder for non-parallel, any-to-many voice conversion. Our Cotatron-based voice conversion system outperforms the previous state-of-the-art method, Blow [12], in terms of both naturalness and speaker similarity scores on a user study, when trained and evaluated with 108 speakers from VCTK dataset [19].

2. Approach

2.1. Speaker-independent linguistic features from TTS

Cotatron is guided with a transcription to extract speaker-independent linguistic features from the speech. Cotatron’s basic architecture is identical to multispeaker Tacotron2 [17, 18]; it jointly learns to align and predict next mel frame from the text encoding, previous mel frame, and the speaker representation:

$$\hat{M}_{1:i}, A_i = \text{Decoder}_{\text{TTS}} \left(\text{Encoder}_{\text{text}}(T), M_{0:i-1}, z^{id} \right), \quad (1)$$

where T, M, A, z^{id} corresponds to text, log mel spectrogram, alignment, and the speaker representation, respectively.

After training, a simple yet effective trick is applied. An alignment A between the speech and the transcription is ob-

tained via feeding all frames of the mel spectrogram into Cotatron with teacher-forcing applied. Then, the speaker-independent linguistic features of the speech are obtained from a matrix multiplication of the alignment and text encoding as Fig. 1:

$$L = \text{matmul} \left(A, \text{Encoder}(T) \right)_{\text{text}}. \quad (2)$$

Per definition, the text encoding contains no speaker information. Besides, the text-audio alignment A is a set of scalar coefficients for weighted summation over encoder timesteps of the text encoding. Hence, we may argue that the Cotatron features L do not explicitly contain a source speaker’s information. We show the degree of speaker disentanglement at Sec. 4.3.

Cotatron features are naturally adequate for synthesizing speech from a large number of speakers; the features can be interpreted as context vectors for Tacotron2’s attention mechanism, which are already optimized for multispeaker speech synthesis. We further expand the coverage of source speakers into arbitrary by replacing the embedding table into an encoder for speaker representation z^{id} . The speaker encoder is composed of 6 layers of 2D CNN, following the reference encoder architecture from Skerry-Ryan *et al.* [20]. Each layer had 3×3 kernel, 2×2 stride with 32, 32, 64, 64, 128, 128 channels. The CNN output is flattened and passed through a 256-unit GRU to obtain the fixed-length speaker representation from the final state.

2.2. Voice conversion

2.2.1. Residual Encoder

Let’s consider a decoder reconstructing the speech from Cotatron features. Even when the rhythm of the transcription is given via Cotatron features, other components of the speech may vary. For example, the intonation may vary within the speech of the same text with identical rhythm. It is therefore insufficient for the decoder to use only the Cotatron features and the speaker representation. To fill the gap of information, we design an encoder to provide decoder a residual feature R .

The residual encoder (Fig. 2b) is built with 6 layers of 2D CNN as the speaker encoder did, but strides are not applied across time to preserve the temporal dimension of the mel spectrogram. Each layer had 3×3 kernel with 2×1 stride and 32, 32, 64, 64, 128, 128 channels. If the dimension of the residual features is too wide, the residual encoder may learn to cheat by encoding the information that is related to the individual speaker – *e.g.*, absolute pitch. We find that a single-channeled output helps to prevent the residual features from containing characteristic of the individual speaker, and is enough to represent residual information of the speech; this approach was also used by Lian *et al.* [4]. After projecting to a single channel, instance normalization [21] is applied to prevent the residual representation from containing speaker-dependent information. Finally, the values are smoothed after tanh activation by applying convolution with a Hann function of window size 21.

2.2.2. VC Decoder

The decoder for voice conversion (Fig. 2c) is trained to reconstruct the mel spectrogram from a given pair of information; the Cotatron features L and the residual feature R are concatenated channel-wise, and then conditioned with 256-dimension speaker embedding y^{id} retrieved from a lookup table as:

$$M_{s \rightarrow *} = \text{Decoder}_{\text{vc}} \left(\text{concat} (L_s, R_s), y_*^{id} \right). \quad (3)$$

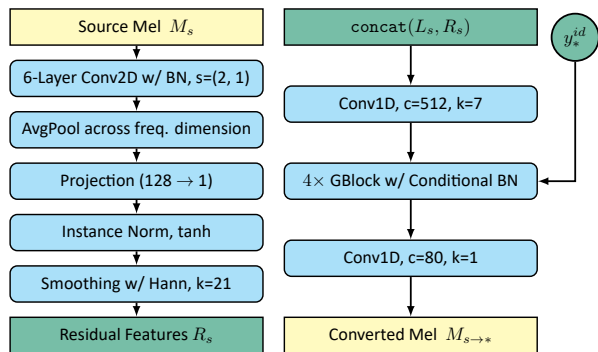
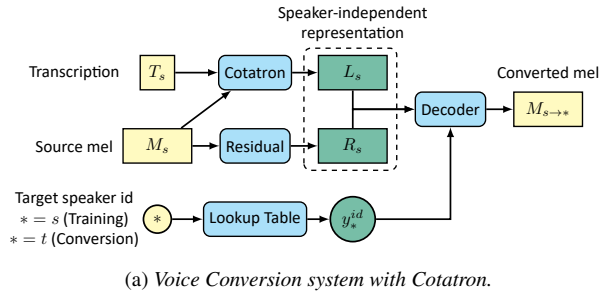


Figure 2: Network architectures. s, c, k denotes the stride, number of channels, kernel size of convolution layer, respectively. Speaker representation y^{id} conditions the VC decoder via conditional batch normalization layer within residual blocks. Refer to Binkowski *et al.* [22] for the detailed architecture of GBlock.

The asterisk symbol can be either s or t , each representing source/target of the voice conversion. Thus, $M_{s \rightarrow s}$ denotes reconstruction, and $M_{s \rightarrow t}$ denotes voice conversion from s to t .

Following the model architecture of GAN-TTS [22], the VC decoder is constructed with a stack of four GBlocks without upsampling. Each GBlock has 512, 384, 256, 192 channels, respectively. For speaker conditioning, the embedding of the target speaker y^{id} is injected via a conditional batch normalization layer [23] within the GBlocks, after an affine transformation. We empirically observed that concatenation of speaker embedding leads to worse results. Neither the hyper-conditioning [24] nor the weight demodulation [25] did not help. There might be room for improvement in design choices of decoder architecture, but we leave it as a future work since it is beyond the scope of this work.

Note that the VC decoder is only trained to reconstruct the mel spectrogram with representations from the identical speaker. Though it is possible to directly train the conversion in an adversarial manner, we show the effectiveness of Cotatron on voice conversion using only reconstruction loss.

3. Experimental setup

3.1. Dataset

Our voice conversion system is trained and evaluated with a VCTK dataset [19], which consists of 46 hours of English speech from 108 speakers. Similar to what Blow had done [12], we split the data into train, validation, and test splits by randomly selecting 80%, 10%, 10% of the data, respectively. To

prevent overlap of transcription between data splits, the data is split with respect to their transcription, not the number of files.

To stabilize the training of multispeaker TTS, we incorporate a subset of LibriTTS [26], which is a dataset specialized in training TTS systems. Speakers with more than 5 minutes of speech are chosen from LibriTTS’ *train-clean-100* subset.

All audios longer than 10 seconds are not used for training to allow efficient batching. The audios are resampled to sampling rate 22.05 kHz and then normalized without silence removal. The statistics of the dataset are shown in Table 1.

Table 1: *Dataset statistics. For LibriTTS train-clean-100 split, speakers with less than 5 minutes of speech are removed.*

Dataset	# speakers	Length (h)
VCTK [19] train / val / test	108	34.6 / 4.5 / 4.2
LibriTTS [26]		
<i>train-clean-100</i>	123	23.4
<i>dev-clean</i>	40	9.0
<i>test-clean</i>	39	8.6

3.2. Training

3.2.1. Cotatron

Cotatron is trained with the aforementioned subset of LibriTTS, which is based on the *train-clean-100* split. Then, the model is transferred to learn with both LibriTTS and VCTK train split. To enhance the stability of text-audio alignment learning, the autoregressive decoder is teacher-forced with a rate of 0.5, *i.e.*, input mel frame is randomly selected from either ground truth frame or previously generated frame. Furthermore, we find it helpful to train extra MLP with dropout for speaker classification on top of z^{id} from the speaker encoder, using cross-entropy loss \mathcal{L}_{id} . Overall, Cotatron is trained with the sum of mel spectrogram reconstruction loss and speaker classification loss:

$$\mathcal{L}_{\text{cotatron}} = \left\| \hat{M}_{s,pre} - M_s \right\|_2^2 + \left\| \hat{M}_{s,post} - M_s \right\|_2^2 + \mathcal{L}_{id}, \quad (4)$$

where $\hat{M}_{s,pre}$ and $\hat{M}_{s,post}$ denote output before and after the Cotatron’s post-net [17], respectively.

Throughout the training process, Adam optimizer [27] is used with batch size 64. The initial learning rate 3×10^{-4} is used for the first 25k steps and then exponentially decayed to 1.5×10^{-5} for the next 25k steps. After the model converges with LibriTTS, we add VCTK and reuse the learning rate decay scheme. Weight decay of 1×10^{-6} is used for Adam optimizer, and the gradient is clipped to 1.0 to prevent gradient explosion.

3.2.2. Mel-Spectrogram Reconstruction

After the training of Cotatron, the components for voice conversion system is trained on top of Cotatron features. The residual encoder and the VC decoder is jointly trained with mel spectrogram reconstruction loss:

$$\mathcal{L}_{vc} = \|M_{s \rightarrow s} - M_s\|_2^2. \quad (5)$$

During the reconstruction training phase, Cotatron is set to evaluation mode; all dropout layers are turned off, and the autoregressive decoder is always teacher-forced to provide consistent features for VC decoder. Adam optimizer with constant learning rate 3×10^{-4} is used with weight decay 1×10^{-6} and batch size 128. Gradient clipping is not used here.

3.3. Conversion

To convert one voice to another, we first extract the speaker-independent features, L_s, R_s , from the source speech with Cotatron and residual encoder, respectively. Then, the embedding of the target speaker y_t^{id} is retrieved from the lookup table. Finally, a pair of speaker-independent features and target speaker embedding is used to produce a converted mel spectrogram, $M_{s \rightarrow t}$. The resulting mel spectrogram is then inverted into raw audio using MelGAN [28], which is trained with LibriTTS train split and then fine-tuned with the entire VCTK dataset.

3.4. Implementation details

For robust alignment stability against length variation, we apply the Dynamic Convolution Attention (DCA) mechanism [29]. The speaker representation is extracted from the ground-truth mel spectrogram with the speaker encoder, and then repeatedly concatenated with text encoder output to feed the autoregressive decoder of Cotatron. For both Cotatron and the voice conversion system, the training data is augmented with representation mixing [30], *i.e.*, graphemes are randomly replaced with phonemes if the word is available in CMUdict [31]. Both decoders produce 80-bin log mel spectrogram, which is computed from 22.05 kHz raw audio using STFT with window size 1024, hop size 256, Hann window, and a mel filterbank spanning from 70 Hz to 8000 Hz. The voice conversion systems are implemented with PyTorch [32] and trained for 10 days with two NVIDIA V100 (32GB) GPU using data parallelism.

3.5. Evaluation metrics

We validate the effectiveness of our method with both subjective and objective metrics, using 100 and 10,000 audio samples per each measurement, respectively.

Mean Opinion Score (MOS). To assess the naturalness of converted speech, we measure the mean opinion score (MOS) on a 5-point scale at Amazon Mechanical Turk (MTurk). A total of 100 audio samples are generated for each case with a random pair of source speech and target speaker, which contains all possible gender combinations. The audio samples from our method and natural speech are downsampled to rate 16 kHz to match the results from Blow [12]. Each sample is assigned to 5 human listeners, and the highest/lowest score is discarded.

Degradation Mean Opinion Score (DMOS). Another user study is done to assess speaker similarity between the converted speech and the target speaker’s original recording. The degradation mean opinion score (DMOS) on a 5-point scale is measured at MTurk with the same settings from the MOS experiment.

Speaker Classification Accuracy (SCA). Our system should be able to fool the speaker classifier as if the converted speech was spoken from the target speaker. The speaker classifier is an MFCC-based single-layer classifier, which is identical to the one used with Blow [12] for a fair comparison. The classifier is trained with 108 speakers from the VCTK train split and achieved 99.4% top-1 accuracy on the test split. The MFCC is directly calculated from the log mel spectrogram if possible.

Voicing Decision Error (VDE). As a proxy metric for content consistency between source and converted speech, we measure the rate of voicing decision match between them, adapting a metric of end-to-end prosody transfer for speech synthesis [20]. The voicing decision is obtained via rVAD [33] with a VAD threshold value set to 0.7.

4. Results

4.1. Many-to-many conversion

We compare our system with Blow [12], which is the only literature to date on many-to-many voice conversion with all speakers of VCTK. As presented in Table 2, our system shows significantly better results on both MOS and DMOS than Blow, even when only the Cotatron features are used without residual features. Incorporating the residual encoder on our system has further enhanced the MOS. It should, however, be noted that the objective results on speaker similarity (SCA) are contradicting that from the subjective results (DMOS). Future work should, therefore, revisit and establish objective speaker similarity metrics for voice conversion systems.

Table 2: Results of many-to-many voice conversion.

Approach	MOS	DMOS	SCA
Source as target	4.28 ± 0.11	1.71 ± 0.22	0.9%
Target as target	4.28 ± 0.11	4.78 ± 0.08	99.4%
Blow	2.41 ± 0.14	1.95 ± 0.16	86.8%
Cotatron (ours)			
w/o residual	3.18 ± 0.14	4.06 ± 0.17	73.3%
full model	3.41 ± 0.14	3.89 ± 0.18	78.5%

4.2. Any-to-many conversion and the use of ASR

Considering the technical demands of the real-world applications, we further explore the generalization power of our voice conversion system. First, we consider any-to-many setting – *i.e.*, converting arbitrary speakers’ speech to that of speakers that are seen during training. Next, we inspect the reliability of using ASR transcription, which enables a fully automatic pipeline of our system without manual transcription. For any-to-many conversion experiment, we randomly sample speeches from LibriTTS *test-clean* split and convert them into speakers of VCTK. For ASR, wav2letter++ [34, 35] is used.

In Table 3, we present the MOS, SCA, and VDE for all possible cases of input. First, all of the MOS results are much better than the previous method in Table 2, though the scores from any-to-many setting are slightly lower than that of many-to-many setting. Next, the differences of SCA across the cases are negligible, and the values of VDE are minimal when considering the accuracy of the VAD module. These results suggest that the conversion quality is rather unaffected by using (1) source speech from speakers that are unseen during training, and/or (2) automated transcription from ASR. Besides, it is surprising to observe that the word errors of automated transcription do not damage the performance; this would seem to suggest that most of the transcription errors originate from their homophones, *e.g.*, *site* is often wrongly transcribed as *sight*.

4.3. Degree of disentanglement

To quantify the degree of speaker disentanglement of features from Cotatron and the residual encoder, we additionally train a neural network for classifying speakers from the VCTK dataset with a given set of features. In the case of ideal speaker disentanglement, the SCA will be close to that of random guessing: 0.9%. Each classification network is built with 4 layers of 1D CNN and batch normalization, followed by the temporal max-pooling layer and MLP with dropout.

Table 3: Results of any-to-many conversion and using ASR transcription. The values are expected to be similar across the rows.

Input Transcription	MOS	SCA	VDE
<i>VCTK test</i> → <i>VCTK test (many-to-many)</i>			
1-a. ground truth	3.41 ± 0.14	78.5%	2.98%
1-b. ASR (WER 12.6%)	3.44 ± 0.12	77.8%	3.03%
<i>LibriTTS test-clean</i> → <i>VCTK test (any-to-many)</i>			
2-a. ground truth	2.84 ± 0.14	73.6%	11.9%
2-b. ASR (WER 7.0%)	2.83 ± 0.15	71.7%	11.7%

As shown in Table 4, the SCA with Cotatron features and the residual features are significantly lower than that from the source mel spectrogram. These results indicate that our method effectively disentangles the speaker’s identity from the speech, while it is noteworthy to mention that the network was slightly able to guess the speaker using only Cotatron features.

Table 4: Degree of speaker disentanglement.

Input Feature	Random	L_s	(L_s, R_s)	M_s
SCA	0.9%	35.2%	54.0%	97.9%

5. Discussion

In this paper, we proposed Cotatron, a transcription-guided speech encoder for speaker-independent linguistic representation, which is based on the multispeaker Tacotron2 architecture. Our Cotatron-based voice conversion system reaches state-of-the-art performance in terms of both naturalness and speaker similarity on conversion across 108 speakers from the VCTK dataset and shows promising results on conversion from arbitrary speakers. Even when the automated transcription with errors is fed, the performances remained the same.

To our best knowledge, Cotatron is the first model to encode the speaker-independent linguistic representation by explicitly aligning the transcription with given speech. This could open a new path towards multi-modal approaches for speech processing tasks, where only the speech modality was usually being used. For example, one may consider training a transcription-guided speech enhancement system based on Cotatron features. Furthermore, traditional speech features that were utilized for lip motion synthesis can be possibly replaced with Cotatron features to incorporate the transcription for better quality.

Still, there is plenty of room for improvement in the voice conversion system with Cotatron. Despite our careful design choices, the residual encoder seems to provide speech features that are entangled with speaker identity, which may harm the conversion quality or even cause mispronunciation issues. Besides, methods for conditioning the target speaker’s representation could be possibly changed; *e.g.*, utilizing a pre-trained speaker verification network as a speaker encoder may enable any-to-any conversion with our system.

6. Acknowledgments

The authors would like to thank Gaku Kotani from U. Tokyo, June Young Yi, and Junhyeok Lee from MindsLab Inc., and other reviewers who elected to remain anonymous for providing beneficial feedback on the initial version of this paper.

7. References

- [1] L. Sun, K. Li, H. Wang, S. Kang, and H. Meng, "Phonetic posteriorgrams for many-to-one voice conversion without parallel data training," in *2016 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2016, pp. 1–6.
- [2] Y. Saito, Y. Ijima, K. Nishida, and S. Takamichi, "Non-parallel voice conversion using variational autoencoders conditioned by phonetic posteriorgrams and d-vectors," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5274–5278.
- [3] C.-c. Yeh, P.-c. Hsu, J.-c. Chou, H.-y. Lee, and L.-s. Lee, "Rhythm-flexible voice conversion without parallel data using cycle-gan over phoneme posteriorgram sequences," in *2018 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2018, pp. 274–281.
- [4] Z. Lian, J. Tao, Z. Wen, B. Liu, Y. Zheng, and R. Zhong, "Towards fine-grained prosody control for voice conversion," *arXiv preprint arXiv:1910.11269*, 2019.
- [5] H. Lu, Z. Wu, D. Dai, R. Li, S. Kang, J. Jia, and H. Meng, "One-shot voice conversion with global speaker embeddings," *Proc. Interspeech 2019*, pp. 669–673, 2019.
- [6] K. Qian, Y. Zhang, S. Chang, X. Yang, and M. Hasegawa-Johnson, "AutoVC: Zero-shot voice style transfer with only autoencoder loss," in *Proceedings of the 36th International Conference on Machine Learning*, vol. 97. PMLR, 2019, pp. 5210–5219.
- [7] J.-c. Chou, C.-c. Yeh, H.-y. Lee, and L.-s. Lee, "Multi-target voice conversion without parallel data by adversarially learning disentangled audio representations," *arXiv preprint arXiv:1804.02812*, 2018.
- [8] J.-c. Chou, C.-c. Yeh, and H.-y. Lee, "One-shot voice conversion by separating speaker and content representations with instance normalization," *arXiv preprint arXiv:1904.05742*, 2019.
- [9] A. van den Oord, O. Vinyals *et al.*, "Neural discrete representation learning," in *Advances in Neural Information Processing Systems*, 2017, pp. 6306–6315.
- [10] S. Ding and R. Gutierrez-Osuna, "Group latent embedding for vector quantized variational autoencoder in non-parallel voice conversion," *Proc. Interspeech 2019*, pp. 724–728, 2019.
- [11] A. T. Liu, P.-c. Hsu, and H.-y. Lee, "Unsupervised end-to-end learning of discrete linguistic units for voice conversion," *Proc. Interspeech 2019*, pp. 1108–1112, 2019.
- [12] J. Serrà, S. Pascual, and C. S. Peralès, "Blow: a single-scale hyperconditioned flow for non-parallel raw-audio voice conversion," in *Advances in Neural Information Processing Systems*, 2019, pp. 6790–6800.
- [13] J. Zhang, Z. Ling, Y. Jiang, L. Liu, C. Liang, and L. Dai, "Improving sequence-to-sequence voice conversion by adding text-supervision," in *2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 6785–6789.
- [14] J. Zhang, Z. Ling, and L.-R. Dai, "Non-parallel sequence-to-sequence voice conversion with disentangled linguistic and speaker representations," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2019.
- [15] F. Biadsy, R. J. Weiss, P. J. Moreno, D. Kanvesky, and Y. Jia, "Parrottron: An end-to-end speech-to-speech conversion model and its applications to hearing-impaired speech and speech separation," *arXiv preprint arXiv:1904.04169*, 2019.
- [16] M. Zhang, X. Wang, F. Fang, H. Li, and J. Yamagishi, "Joint training framework for text-to-speech and voice conversion using multi-source tacotron and wavenet," *Proc. Interspeech 2019*, pp. 1298–1302, 2019.
- [17] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerry-Ryan *et al.*, "Natural tts synthesis by conditioning wavenet on mel spectrogram predictions," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4779–4783.
- [18] Y. Jia, Y. Zhang, R. Weiss, Q. Wang, J. Shen, F. Ren, P. Nguyen, R. Pang, I. L. Moreno, Y. Wu *et al.*, "Transfer learning from speaker verification to multispeaker text-to-speech synthesis," in *Advances in neural information processing systems*, 2018, pp. 4480–4490.
- [19] J. Yamagishi, C. Veaux, K. MacDonald *et al.*, "Cstr vctk corpus: English multi-speaker corpus for cstr voice cloning toolkit (version 0.92)," 2019.
- [20] 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 *International Conference on Machine Learning*, 2018, pp. 4693–4702.
- [21] D. Ulyanov, A. Vedaldi, and V. S. Lempitsky, "Improved texture networks: Maximizing quality and diversity in feed-forward stylization and texture synthesis," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4105–4113.
- [22] M. Bińkowski, J. Donahue, S. Dieleman, A. Clark, E. Elsen, N. Casagrande, L. C. Cobo, and K. Simonyan, "High fidelity speech synthesis with adversarial networks," in *International Conference on Learning Representations (ICLR)*, 2020.
- [23] V. Dumoulin, J. Shlens, and M. Kudlur, "A learned representation for artistic style," in *International Conference on Learning Representations (ICLR)*, 2017.
- [24] D. Ha, A. Dai, and Q. V. Le, "Hypernetworks," in *International Conference on Learning Representations (ICLR)*, 2017.
- [25] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, "Analyzing and improving the image quality of stylegan," *arXiv preprint arXiv:1912.04958*, 2019.
- [26] H. Zen, R. Clark, R. J. Weiss, V. Dang, Y. Jia, Y. Wu, Y. Zhang, and Z. Chen, "Libritts: A corpus derived from librispeech for text-to-speech," *Proc. Interspeech 2019*, pp. 1526–1530, 2019.
- [27] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *International Conference on Learning Representations (ICLR)*, 2015.
- [28] K. Kumar, R. Kumar, T. de Boissiere, L. Gestein, W. Z. Teoh, J. Sotelo, A. de Brébisson, Y. Bengio, and A. C. Courville, "Melgan: Generative adversarial networks for conditional waveform synthesis," in *Advances in Neural Information Processing Systems*, 2019, pp. 14 881–14 892.
- [29] E. Battenberg, R. Skerry-Ryan, S. Mariooryad, D. Stanton, D. Kao, M. Shannon, and T. Bagby, "Location-relative attention mechanisms for robust long-form speech synthesis," *arXiv preprint arXiv:1910.10288*, 2019.
- [30] K. Kastner, J. F. Santos, Y. Bengio, and A. Courville, "Representation mixing for tts synthesis," in *2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 5906–5910.
- [31] R. L. Weide, "The cmu pronouncing dictionary," URL: <http://www.speech.cs.cmu.edu/cgibin/cmudict>, 1998.
- [32] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems*, 2019, pp. 8024–8035.
- [33] Z.-H. Tan, N. Dehak *et al.*, "rvad: An unsupervised segment-based robust voice activity detection method," *Computer Speech & Language*, vol. 59, pp. 1–21, 2020.
- [34] N. Zeghidour, Q. Xu, V. Liptchinsky, N. Usunier, G. Synnaeve, and R. Collobert, "Fully convolutional speech recognition," *arXiv preprint arXiv:1812.06864*, 2018.
- [35] V. Pratap, A. Hannun, Q. Xu, J. Cai, J. Kahn, G. Synnaeve, V. Liptchinsky, and R. Collobert, "Wav2letter++: A fast open-source speech recognition system," in *2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 6460–6464.