PROSODYSPEECH: TOWARDS ADVANCED PROSODY MODEL FOR NEURAL TEXT-TO-SPEECH

Yuanhao Yi, Lei He, Shifeng Pan, Xi Wang, Yujia Xiao

Microsoft

ABSTRACT

This paper proposes ProsodySpeech, a novel prosody model to enhance encoder-decoder neural Text-To-Speech (TTS), to generate high expressive and personalized speech even with very limited training data. First, we use a Prosody Extractor built from a large speech corpus with various speakers to generate a set of prosody exemplars from multiple reference speeches, in which Mutual Information based Style content separation (MIST) is adopted to alleviate ”content leakage” problem. Second, we use a Prosody Distributor to make a soft selection of appropriate prosody exemplars in phone-level with the help of an attention mechanism. The resulting prosody feature is then aggregated into the output of text encoder, together with additional phone-level pitch feature to enrich the prosody. We apply this method into two tasks: highly expressive multi style/emotion TTS and few-shot personalized TTS. The experiments show the proposed model outperforms baseline FastSpeech 2 + GST with significant improvements in terms of similarity and style expression.

Index Terms— TTS, Prosody, MIST, Attention, Few-shot

1. INTRODUCTION

Neural TTS models have rapidly improved with various neural networks, such as Deep Neural Network (DNN) [1] and Recurrent Neural Network with Long-Short Term Memory (LSTM-RNN) [2], have been applied to speech synthesis and demonstrated their superiority over traditional Hidden Markov Models (HMMs) [3]. Recently, Some new neural models, such as Tacotron [4], Tacotron 2 [5], Deep voice 3 [6], Clarinet [7], TransformerTTS [8] has been proposed to generate speech autoregressively from text input that can achieve performance very close to human quality. In order to increase inference speed and generate more robust speech, non-autoregressive TTS models such as FastSpeech [9] and FastSpeech 2 [10] are proposed with robust and fast parallel generation.

However, there are still challenges for neural TTS to reconstruct ”vivid” prosody when data includes diverse style distribution or even limited data amount, regarding the prosody learning is not instructive and effective enough among the end-to-end model parameters. Recent approaches utilize deep learning to generate a unsupervised learned single latent style embedding [11, 12, 13, 14, 15, 16, 17] or combine with other explicit variables such as pitch and rhythm [18, 19, 20] for prosody modeling. But, the discrepancy between reference and target speech on length or content defects the effectiveness of single style embedding working during the inference stage, which is hard to select appropriate reference audio in real application. Attentron [21] proposes a novel architecture to utilize multiple reference audios used by two speech encoders, fine-grained encoder and coarse-grained encoder, but the fine-grained encoder is not friendly to calculate attention on a massive frame-level style embedding set.

To that end, we propose ProsodySpeech, a neural TTS framework based on FastSpeech 2 with additional Prosody Extractor and Prosody Distributor modules for better prosody modeling and transfer. The contributions of this paper are summarized as follows:

- We present a well pre-trained Prosody Extractor with the help of mutual information estimator that learn prosody representations disentangled from phonetic content.
- We present a novel Prosody Distributor with the help of an attention mechanism to make a soft selection of appropriate prosody exemplars to get phone-level prosody representations.

The rest of this paper is organized as follows: Details of the proposed model are presented in Section 2. Implementation details are illustrated in Section 3. Experimental results are shown in Section 4 and Section 5 concludes this paper.

2. MODEL ARCHITECTURE

2.1. Encoder and Decoder

As illustrate in Figure 1 both Encoder and Decoder are composed of 6 Feed-Forward Transformer (FFT) blocks [22], with conditional layer normalization [23] leveraged by AdaSpeech [24] that replace the scale and bias generated by two linear layers respectively from 128-dimensional speaker embedding, to decide whether only adapt partial parameters for fine-tuning. The number of attention heads is set

Audio examples at: https://luckeryi.github.io/microsoft/ProsodySpeech

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

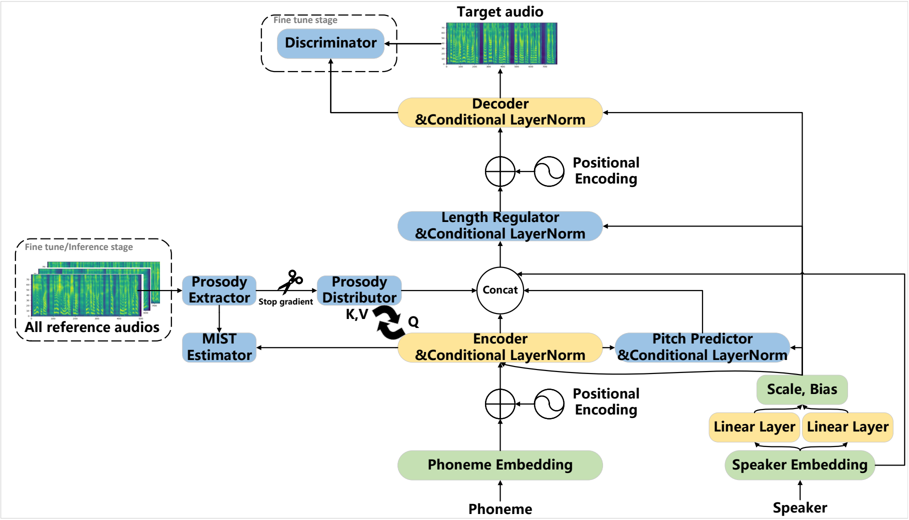

Fig. 1. The overall architecture for ProsodySpeech.

to 4 and the attention dimension is 384. The Feed-Forward layers have 384 and 1536 hidden units respectively. The Encoder takes 512-dimensional phoneme embedding followed by Positional Encoding [22] as input to output phone-level contextual feature $\mathbf{F}_{\text{context}}$. Decoder takes the upsampled frame-level $\mathbf{F}_{\text{context}}$, concatenating with expanded prosody feature $\mathbf{F}_{\text{prosody}}$, pitch feature $\mathbf{F}_{\text{pitch}}$ and speaker feature $\mathbf{F}_{\text{speaker}}$ as input to generate Mel-Spectrogram.

2.2. Length Regulator and Pitch Predictor

Both Length Regulator and Pitch Predictor are composed of 2 layers of 1D convolutional network with 384 filters with shape of 1×3, followed by ReLU activation. And each layer is followed by conditional layer normalization with dropout 0.1. We use ground truth phoneme duration to expand the phone-level $\mathbf{F}_{\text{context}}$ into frame-level. And we use predict phone-averaged $\mathbf{F}_{\text{pitch}}$ with a Tanh activation to integrate with $\mathbf{F}_{\text{context}}$ during training.

2.3. Prosody Extractor and MIST Estimator

Prosody Extractor has similar structure with GST [11] to generate sentence level style latent representation, and we replace the reference encoder to 4 aforementioned FFT blocks. The MIST Estimator is designed to alleviate ”content leakage” problem when the reference speech does not match the content input, the output may not contain all of the content of the input text, and we leverage the method of [25, 26] that constructs a lower bound of mutual information based on Donsker-Varadhan representation of KL divergence [27]:

$$I(Y, Z) \geq \widehat{I}_T(Y, Z), \tag{1}$$

$$\widehat{I}_T(Y, Z) = \underset{M}{sup}\, E_{P_{Y,Z}}[M] - log(E_{P_Y * P_Z}[e^M]) \tag{2}$$

where M is the MIST Estimator and composed of 2 linear layers with 512 hidden units followed by Elu activation, and 1 linear layer with 1 hidden unit followed by Tanh activation. And this equation has the ability to estimate mutual information between Y and Z by maximizing the lower bound via gradient descent. Thus, this can be a min-max problem where we maximize the lower-bound of the equation above which means training a better MIST Estimator, and minimize the mutual imformation between $\mathbf{F}_{\text{prosody}}$ and $\mathbf{F}_{\text{context}}$. Thus, the training process for MIST Estimator must be alternative that is similar to GAN.

2.4. Prosody Distributor and Discriminator

The Prosody Distributor is composed of a multi-head attention and Figure 1 illustrates how the attention mechanism works. It is designed to take advantage of prosody exemplars set which is generated by the pre-trained Prosody Extractor using multiple reference speeches to enrich the diversity of prosody features integrated with context features as much as possible. The prosody feature $\mathbf{F}_{\text{prosody}}$ is calculated as :

$$\mathbf{F}_{\text{prosody}} = \text{Attention}(\mathbf{F}_{\text{context}}, \mathbf{F}_{\text{prosody\_set}}) \tag{3}$$

where $\mathbf{F}_{\text{prosody\_set}}$ is the prosody exemplars set generated by the pre-trained Prosody Extractor using musing multiple reference speeches. The number of attention heads is set to 4 and the attention dimension is 384.

Expecting to generate a realistic voice, we adopted a adversarial training method which helps the output distribution of predicted mel spetrogram be similar to the target mel spetrogram. We use the same structure of Discriminator network as [28]. During the adversarial training, we also fixed two window sizes of 50 frames and 100 frames to randomly intercept both the predicted and target mel spetrograms as input to the Discriminator at each training step.

3. IMPLEMENTATION

3.1. Pre-training and Loss Function

We use a 3-steps warm-up strategy to train a general source model in order to get a prosody-cleaned Encoder and a stable Prosody Distributor.

First step, like [25], we first train a vanilla model without Prosody Extractor, Prosody Distributor and $\mathbf{F}_{\text{pitch}}$, to get a prosody-cleaned Encoder. The overall loss function $\mathcal{L}_{p1}$ is:

$$\mathcal{L}_{p1} = \mathcal{L}_{\text{recons}} + \lambda_1 \mathcal{L}_{\text{ssim}} + \lambda_2 \mathcal{L}_{\text{dur}} \tag{4}$$

where $\mathcal{L}_{\text{recons}}$ is the loss of reconstruction and we use $\mathcal{L}_1$ + $\mathcal{L}_2$. $\mathcal{L}_{\text{ssim}}$ is the loss of Structural SIMilarity (SSIM), which is widely used to measure the similarity of two images, to help jointly optimize the model. $\mathcal{L}_{\text{dur}}$ is the loss of phone-level duration in the logarithmic domain. We set $\lambda_1 = 3.0$, $\lambda_2 = 1.0$.

Second step, we add Prosody Extractor and $\mathbf{F}_{\text{pitch}}$ to the model, while reloading and fixing parameters of the Encoder from first step. We introduce the MIST Estimator and conduct alternately training to get a prosody-only Prosody Extractor. The overall loss function $\mathcal{L}_{p2}$ is:

$$\begin{aligned} \mathcal{L}_{p2} = \mathcal{L}_{\text{recons}} + \lambda_1 \mathcal{L}_{\text{ssim}} + \lambda_2 \mathcal{L}_{\text{dur}} + \lambda_3 \mathcal{L}_{\text{picth}} \\ + \lambda_4 \underset{T}{max}\{ReLU(\widehat{\mathcal{I}}_{\mathcal{T}}(\mathbf{F}_{\text{prosody}}, \mathbf{F}^{(\text{s})}_{\text{context}}))\} \end{aligned} \tag{5}$$

where $\mathcal{L}_{\text{pitch}}$ is the loss of phone-averaged pitch in log domain. $\mathbf{F}^{(\text{s})}_{\text{context}}$ is a random sample of the context features to compute the mutual information in each training step. And for MIST Estimator training round, we optimize $-\widehat{I}_T(\mathbf{F}_{\text{prosody}}, \mathbf{F}^{(\text{s})}_{\text{context}})$. ReLU activation function is designed to obtain the non-negative mutual information value. $\lambda_1 = 2.0$, $\lambda_2 = 1.0$, $\lambda_3 = 1.0$, $\lambda_4 = 1.0$.

Last step, we add Prosody Extractor, $\mathbf{F}_{\text{pitch}}$ and Prosody Distributor to the model, while reloading and fixing parameters of Encoder, Pitch Predictor and Prosody Extractor, to get a stable Prosody Distributor. The overall loss function $\mathcal{L}_{p3}$ is:

$$\mathcal{L}_{p3} = \mathcal{L}_{\text{recons}} + \lambda_1 \mathcal{L}_{\text{ssim}} + \lambda_1 \mathcal{L}_{\text{dur}} \tag{6}$$

where $\lambda_1 = 1.0$, $\lambda_2 = 1.0$.

3.2. Fine-tuning and Loss Function

During fine-tuning stage, we introduce Discriminator while reloading all parameters from the pre-trained source model and fixing Prosody Extractor and MIST Estimator parameters. We use WGAN-GP [29] to train the Discriminator. The overall loss function $\mathcal{L}_f$ is:

$$\begin{aligned} \mathcal{L}_f = \mathcal{L}_{\text{recons}} + \lambda_1 \mathcal{L}_{\text{ssim}} + \lambda_2 \mathcal{L}_{\text{dur}} + \lambda_3 \mathcal{L}_{\text{picth}} \\ + \lambda_4 \underset{T}{max}\{ReLU(\widehat{\mathcal{I}}_{\mathcal{T}}(\mathbf{F}_{\text{prosody}}, \mathbf{F}^{(\text{s})}_{\text{context}}))\} + \lambda_5 \mathcal{L}_{\text{adv}} \end{aligned} \tag{7}$$

where $\mathcal{L}_{\text{adv}}$ represents adversarial loss and $\lambda_1 = 3.0$, $\lambda_2 = 1.0$, $\lambda_3 = 0.1$, $\lambda_4 = 1.0$, $\lambda_5 = 0.1 \times \mathcal{L}_{\text{recons}}$. For Discriminator training round, we optimize $-\mathcal{L}_{\text{adv}} + \beta \mathcal{L}_{\text{gp}}$, and $\mathcal{L}_{\text{gp}}$ represents gradient penalty regularization, β is set 10.

4. EXPERIMENTS

4.1. Datatset

For highly expressive multi style/emotion TTS task, we train the source model on a proprietary Chinese Audio Book corpus mixed with English of General News for both Baseline and ProsodySpeech. The corpus consists of 40 speakers with multi-style, multi-role and total 382,724 sentences about 447.02 hours. We fine-tune the source model on a Chinese Audio Book corpus and it consists of a middle-aged female with 7 styles (gentle, nervous, happy, complain, strict, angry, sad) and total 3920 sentences about 4.02 hours. For few-shot personalized TTS task, we train the source model on a proprietary English corpus for both Baseline and ProsodySpeech. The corpus consists of 450 speakers with single style and total 701,255 sentences about 753.47 hours. We fine-tune the source model on 2 male and 2 female personalized voices and 15 sentences about 1 minute for each voice.

4.2. System Configuration

we use FastSpeech 2 + GST with conditional layer normalization module as our Baseline. For highly expressive multi style/emotion TTS task, and we additionally add style embedding $\mathbf{F}_{\text{style}}$ and role embedding $\mathbf{F}_{\text{role}}$ both in Baseline and ProsodySpeech. In the inference stage, we use multiple the reference audios with specific style to generate prosody exemplars set when we synthesize the speech with corresponding style in ProsodySpeech. Meanwhile, we use paired content audio as reference audio for Baseline. For few-shot personalized TTS task, we only fine-tune the parameters related to the conditional layer normalization and the speaker embedding while fixing other model parameters both in Baseline and ProsodySpeech. In the inference stage, we use all the reference audios to generate prosody exemplars set when we synthesize the speech in ProsodySpeech. Meanwhile, we randomly select audio from the training data as reference audio for Baseline.

We use the ADAM [30] optimizer with an initial learning rate of 0.5, and a beta value of (0.9, 0.998). We use Noam's learning rate decay scheme [22], warmup steps = 8,000. A MelGAN vocoder [31] was built to reconstruct audio from mel-spectrograms for all models.

4.3. Result

As shown in Figure 2, the first two rows represent sad and angry speech cases synthesized by Baseline and ProsodySpeech, respectively, and ProsodySpeech can model the high frequency harmonics better, and the variation of the fundamental frequency is more abundant and natural. The third row shows the Prosody Distributor's different behaviors (Left: selects nearly the same reference speeches, Right: selects multiple different reference speeches) on each of the last two

Table 1. Style MOS and Similarity MOS (SMOS) on expressive multi style/emotion TTS task.

Metric	Setting	Gentle	Nervous	Happy	Complain	Strict	Angry	Sad
Style MOS	Recording	4.30±0.13	4.46±0.12	4.47±0.11	4.30±0.09	4.25±0.11	4.38±0.13	4.72±0.07
	Baseline	3.58±0.16	3.77±0.15	3.14±0.10	3.36±0.07	3.33±0.08	3.84±0.08	3.71±0.09
	ProsodySpeech	3.57±0.18	3.72±0.16	**3.51±0.09**	**3.69±0.08**	**3.75±0.08**	**3.94±0.08**	**3.92±0.09**
SMOS	Recording	4.37±0.12	4.56±0.13	4.62±0.12	4.59±0.11	4.55±0.17	4.58±0.13	4.61±0.12
	Baseline	3.86±0.07	4.06±0.09	3.78±0.07	4.07±0.10	3.95±0.09	3.99±0.08	3.99±0.08
	ProsodySpeech	**3.98±0.07**	**4.12±0.08**	**4.23±0.07**	**4.38±0.10**	**4.35±0.09**	**4.05±0.08**	**4.22±0.08**

Table 2. SMOS and Comparison MOS (CMOS) on few-shot personalized TTS task.

Metric	Setting	Female1	Female2	Male1	Male2
SMOS	Recording	4.16±0.17	4.33±0.11	4.28±0.10	4.42±0.13
	Baseline	4.08±0.12	3.93±0.08	4.08±0.08	4.02±0.10
	ProsodySpeech	**4.15±0.11**	**4.11±0.07**	**4.26±0.06**	**4.38±0.09**
CMOS	Baseline vs. ProsodySpeech	**0.112**	**0.098**	**0.188**	**0.175**

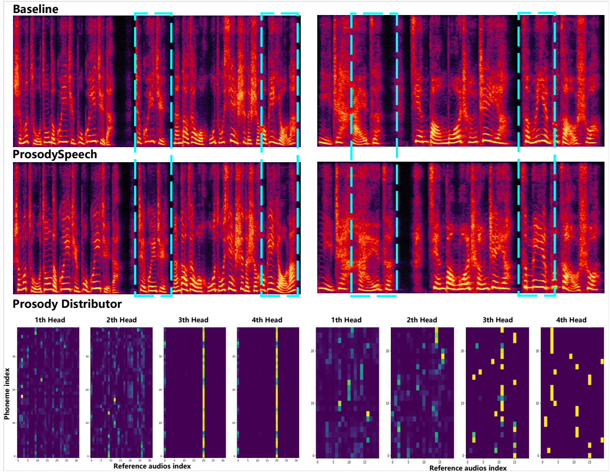

Fig. 2. Linear-spectrograms generated by Baseline and ProsodySpeech using MelGAN vocoder and Prosody Distributor's multi-head attention matrix in ProsodySpeech.

main attention heads. We then evaluate the TTS quality in terms of style expression (how the synthesized voices sound emotionally expressive in this content) and similarity (how the synthesized voices sound similar to reference speaker). Here, human evaluations with Style MOS (Style Mean Opinion Score) are carried out, in which the context (previous and next sentences in paragraph) is presented to judges to better understand the expected appropriate style expression in audio book scenario. While the standard SMOS (Similarity MOS) is adopted for similarity test. Each sentence is listened by 20 crowd-sourcing judges in all tests. For highly expressive multi style/emotion TTS task, we generate 75 sentences for each style, and conduct the Style MOS and SMOS on each style. From Table 1, ProsodySpeech can achieve better style expression and similarity especially on strong-intensity styles(happy, complain, strict, angry, sad). For few-shot personalized TTS task, we generate 45 sentences for each voice, and conduct the SMOS and Comparison MOS (CMOS) which compares the naturalness on each voice, from Table 2, ProsodySpeech can achieve almost the same similarity as recording, and the naturalness is also significantly improved on each voice. We also explore the effectiveness of prosody related modules in ProsodySpeech and Baseline. From Table 3, w/o GST denotes removing GST module, in the same way, F0 denotes $\mathbf{F}_{\text{pitch}}$, MI denotes mutual information constrain and Prosody denotes $\mathbf{F}_{\text{prosody}}$, respectively. All results in performance drop in similarity, demonstrating the effectiveness of each component.

Table 3. SMOS on prosody related modules ablation study.

Setting	SMOS
Recording	4.74±0.06
Baseline	4.51±0.08
- w/o GST	4.38±0.08
ProsodySpeech	**4.61±0.07**
- w/o F0	4.41±0.08
- w/o F0+MI	4.34±0.08
- w/o Prosody	4.38±0.08
- w/o F0+MI+Prosody	4.22±0.07

5. CONCLUSIONS

In this paper, we propose ProsodySpeech, a neural TTS model with advanced prosody modeling to generate high expressive and personalized speech even with very limited training data. ProsodySpeech can significantly improve the similarity and style expression, especially for strong-intensity styles on highly expressive multi style/emotion TTS task. Meanwhile, for few-shot personalized TTS task, ProsodySpeech can also greatly improve the similarity and naturalness.

6. REFERENCES

[1] Heiga Zen and Andrew Senior, "Deep mixture density networks for acoustic modeling in statistical parametric speech synthesis," in *2014 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2014, pp. 3844–3848.

[2] Heiga Zen and Haşim Sak, "Unidirectional long short-term memory recurrent neural network with recurrent output layer for low-latency speech synthesis," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 4470–4474.

[3] Junichi Yamagishi, Bela Usabaev, Simon King, Oliver Watts, John Dines, Jilei Tian, Yong Guan, Rile Hu, Keiichiro Oura, Yi-Jian Wu, et al., "Thousands of voices for hmm-based speech synthesis–analysis and application of tts systems built on various asr corpora," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 5, pp. 984–1004, 2010.

[4] Yuxuan Wang, RJ Skerry-Ryan, Daisy Stanton, Yonghui Wu, Ron J Weiss, Navdeep Jaitly, Zongheng Yang, Ying Xiao, Zhifeng Chen, Samy Bengio, et al., "Tacotron: Towards end-to-end speech synthesis," *arXiv preprint arXiv:1703.10135*, 2017.

[5] Jonathan Shen, Ruoming Pang, Ron J Weiss, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng Chen, Yu Zhang, Yuxuan Wang, Rj Skerrv-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.

[6] Wei Ping, Kainan Peng, Andrew Gibiansky, Sercan O Arik, Ajay Kannan, Sharan Narang, Jonathan Raiman, and John Miller, "Deep voice 3: 2000-speaker neural text-to-speech," *Proc. ICLR*, pp. 214–217, 2018.

[7] Wei Ping, Kainan Peng, and Jitong Chen, "Clarinet: Parallel wave generation in end-to-end text-to-speech," *arXiv preprint arXiv:1807.07281*, 2018.

[8] Naihan Li, Shujie Liu, Yanqing Liu, Sheng Zhao, and Ming Liu, "Neural speech synthesis with transformer network," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019, vol. 33, pp. 6706–6713.

[9] Yi Ren, Yangjun Ruan, Xu Tan, Tao Qin, Sheng Zhao, Zhou Zhao, and Tie-Yan Liu, "Fastspeech: Fast, robust and controllable text to speech," *arXiv preprint arXiv:1905.09263*, 2019.

[10] Yi Ren, Chenxu Hu, Tao Qin, Sheng Zhao, Zhou Zhao, and Tie-Yan Liu, "Fastspeech 2: Fast and high-quality end-to-end text-to-speech," *arXiv preprint arXiv:2006.04558*, 2020.

[11] Yuxuan Wang, Daisy Stanton, Yu Zhang, RJ-Skerry Ryan, Eric Battenberg, Joel Shor, Ying Xiao, Ye Jia, Fei Ren, and Rif A Saurous, "Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis," in *International Conference on Machine Learning*. PMLR, 2018, pp. 5180–5189.

[12] RJ Skerry-Ryan, Eric Battenberg, Ying Xiao, Yuxuan Wang, Daisy Stanton, Joel Shor, Ron Weiss, Rob Clark, and Rif A Saurous, "Towards end-to-end prosody transfer for expressive speech synthesis with tacotron," in *international conference on machine learning*. PMLR, 2018, pp. 4693–4702.

[13] Ya-Jie Zhang, Shifeng Pan, Lei He, and Zhen-Hua Ling, "Learning latent representations for style control and transfer in end-to-end speech synthesis," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 6945–6949.

[14] Wei-Ning Hsu, Yu Zhang, Ron J Weiss, Heiga Zen, Yonghui Wu, Yuxuan Wang, Yuan Cao, Ye Jia, Zhifeng Chen, Jonathan Shen, et al., "Hierarchical generative modeling for controllable speech synthesis," *arXiv preprint arXiv:1810.07217*, 2018.

[15] Eric Battenberg, Soroosh Mariooryad, Daisy Stanton, RJ Skerry-Ryan, Matt Shannon, David Kao, and Tom Bagby, "Effective use of variational embedding capacity in expressive end-to-end speech synthesis," *arXiv preprint arXiv:1906.03402*, 2019.

[16] Guangzhi Sun, Yu Zhang, Ron J Weiss, Yuan Cao, Heiga Zen, and Yonghui Wu, "Fully-hierarchical fine-grained prosody modeling for interpretable speech synthesis," in *ICASSP 2020-2020 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2020, pp. 6264–6268.

[17] Florian Eyben, Sabine Buchholz, Norbert Braunschweiler, Javier Latorre, Vincent Wan, Mark JF Gales, and Kate Knill, "Unsupervised clustering of emotion and voice styles for expressive tts," in *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2012, pp. 4009–4012.

[18] Rafael Valle, Jason Li, Ryan Prenger, and Bryan Catanzaro, "Mellotron: Multispeaker expressive voice synthesis by conditioning on rhythm, pitch and global style tokens," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6189–6193.

[19] Rui Liu, Berrak Sisman, Guang lai Gao, and Haizhou Li, "Expressive tts training with frame and style reconstruction loss," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2021.

[20] Heejin Choi, Sangjun Park, Jinuk Park, and Minsoo Hahn, "Emotional speech synthesis for multi-speaker emotional dataset using wavenet vocoder," in *2019 IEEE International Conference on Consumer Electronics (ICCE)*. IEEE, 2019, pp. 1–2.

[21] Seungwoo Choi, Seungju Han, Dongyoung Kim, and Sungjoo Ha, "Attentron: Few-shot text-to-speech utilizing attention-based variable-length embedding," 2020.

[22] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin, "Attention is all you need," *arXiv preprint arXiv:1706.03762*, 2017.

[23] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016.

[24] Mingjian Chen, Xu Tan, Bohan Li, Yanqing Liu, Tao Qin, Sheng Zhao, and Tie-Yan Liu, "Adaspeech: Adaptive text to speech for custom voice," *arXiv preprint arXiv:2103.00993*, 2021.

[25] Ting-Yao Hu, Ashish Shrivastava, Oncel Tuzel, and Chandra Dhir, "Unsupervised style and content separation by minimizing mutual information for speech synthesis," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 3267–3271.

[26] Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeshwar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and Devon Hjelm, "Mutual information neural estimation," in *International Conference on Machine Learning*. PMLR, 2018, pp. 531–540.

[27] Monroe D Donsker and SR Srinivasa Varadhan, "Asymptotic evaluation of certain markov process expectations for large time. iv," *Communications on Pure and Applied Mathematics*, vol. 36, no. 2, pp. 183–212, 1983.

[28] Juheon Lee, Hyeong-Seok Choi, Chang-Bin Jeon, Junghyun Koo, and Kyogu Lee, "Adversarially trained end-to-end korean singing voice synthesis system," *arXiv preprint arXiv:1908.01919*, 2019.

[29] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville, "Improved training of wasserstein gans," *arXiv preprint arXiv:1704.00028*, 2017.

[30] Diederik P Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[31] Kundan Kumar, Rithesh Kumar, Thibault de Boissiere, Lucas Gestin, Wei Zhen Teoh, Jose Sotelo, Alexandre de Brébisson, Yoshua Bengio, and Aaron Courville, "Melgan: Generative adversarial networks for conditional waveform synthesis," *arXiv preprint arXiv:1910.06711*, 2019.