STYLER: Style Factor Modeling with Rapidity and Robustness via Speech Decomposition for Expressive and Controllable Neural Text to Speech

Keon Lee, Kyumin Park, Daeyoung Kim

School of Computing, KAIST, Daejeon, Republic of Korea

{keon.lee, pkm9403, kimd}@kaist.ac.kr

Abstract

Previous works on neural text-to-speech (TTS) have been addressed on limited speed in training and inference time, robustness for difficult synthesis conditions, expressiveness, and controllability. Although several approaches resolve some limitations, there has been no attempt to solve all weaknesses at once. In this paper, we propose STYLER, an expressive and controllable TTS framework with high-speed and robust synthesis. Our novel audio-text aligning method called Mel Calibrator and excluding autoregressive decoding enable rapid training and inference and robust synthesis on unseen data. Also, disentangled style factor modeling under supervision enlarges the controllability in synthesizing process leading to expressive TTS. On top of it, a novel noise modeling pipeline using domain adversarial training and Residual Decoding empowers noise-robust style transfer, decomposing the noise without any additional label. Various experiments demonstrate that STYLER is more effective in speed and robustness than expressive TTS with autoregressive decoding and more expressive and controllable than reading style non-autoregressive TTS. Synthesis samples and experiment results are provided via our demo page[1], and code is available publicly[2].

Index Terms: neural text-to-speech, style modeling, style transfer, expressive and controllable text-to-speech

1. Introduction

Despite remarkable improvement in neural text-to-speech (TTS) towards human-level reading performance [1, 2, 3, 4], criticism has been raised for the lack of expressiveness and controllability. Since the style of synthesized speech is determined from the average style of speech in the training dataset [5, 6], the TTS models have been limited to represent expressive voice. In order to resolve such a problem, several approaches [7, 8, 9, 10, 11] has been explored so that they can synthesize and control speech with various style.

However, the expressive TTS models have weaknesses in speed and robustness due to autoregressive architecture [12]. Since each frame needs to iterate of all previous time steps to be predicted, decoding consumes significant training and inference overhead, and collapse in one step may fall into the failure of total synthesis [13, 14]. Also, their unsupervised style modeling reports difficult training and has a fundamental limitation on disentangling features [15].

Meanwhile, speech synthesis frameworks without autoregressive decoding have been proposed recently [12, 16, 17]. Utilizing Transformer [18], some non-autoregressive TTS models use self-attention block for parallel decoding [12, 17]. The substitution of autoregressive decoder into duration prediction with supervision shows faster speed and enhances stability. Predicting value only from the single text input, however, these models still have limitations on low expressiveness and weak controllability where the influence of audio decreases.

Even with several advances in previous works, none of them covered all criticized problems at once: speed, robustness, expressivity, and controllability. To achieve this, the following conditions should be satisfied. First, autoregressive architecture needs to be avoided, which induces speed and robustness weaknesses. Second, style factor modeling from source audio should be introduced to enable more expressive and controllable synthesis, enjoying the opportunity of input variants in addition to a single text.

In this paper, we propose *STYLER*, a fast and robust style modeling TTS framework for expressive and controllable speech synthesis. Upon non-autoregressive decoding, we construct style factor modeling to express and control each style factor. In addition to text input, STYLER takes and splits source audio into five components: duration, pitch, energy, content, and noise. Inspired by speech decomposition via information bottleneck [19, 20] in recent voice conversion task, our model separately encodes each style factor and decodes them into speech. Furthermore, our novel noise modeling decomposes noise from reference speech with no labels while successfully encoding the speech style from noisy input. To the best of our knowledge, STYLER is the first approach of TTS framework equipped with rapidity, robustness, expressivity, and controllability at the same time with high naturalness.

The contributions of STYLER are as follows.

- STYLER reveals faster and more robust synthesis in unseen data with a novel audio-text aligning compare to the expressive TTS with the autoregressive decoding.

- STYLER achieves high expressivity and controllability through disentangled style factors with the same level of supervision compare to the non-autoregressive TTS.

- STYLER is also noise-robust, proving that our novel noise modeling pipeline can decompose the noise as other style factors without any additional label.

2. Method

In this section, we describe our model first in an abstract level, then in detail. The noise modeling is introduced in the tail.

2.1. Supervised Speech Decomposition

There are two primary considerations to achieve our goal, non-autoregressive TTS with style factor modeling. First, unlike the voice conversion task, the output speech's content is from text, and it may not be matched to the audio content. Second, recent non-autoregressive TTS frameworks benefit from supervision, while the speech decomposition is performed in unsupervised
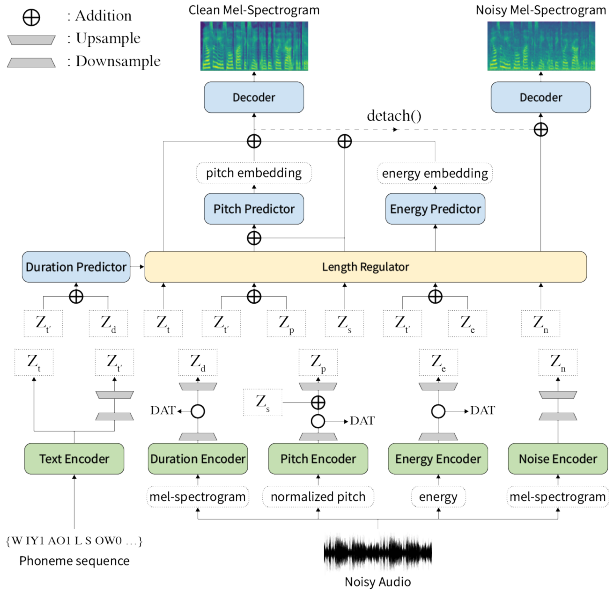
---

[1] https://keonlee9420.github.io/STYLER-Demo/
[2] https://github.com/keonlee9420/STYLER

Figure 1: *An architecture of the proposed model.* $\mathbf{Z}_t$, $\mathbf{Z}_{t'}$, $\mathbf{Z}_d$, $\mathbf{Z}_p$, $\mathbf{Z}_s$, $\mathbf{Z}_e$ and $\mathbf{Z}_n$ *refer to text, downsampled text, duration, pitch, speaker, energy, and noise encoding, and DAT stands for Domain Adversarial Training.*

conditions. Solving these issues, we propose Mel Calibrator and a reinterpretation of speech decomposition under supervision.

2.1.1. Mel Calibrator

Mismatch of length in text and audio has been solved by attention mechanism [10, 11]. Attention, however, has been claimed to be unstable, especially when synthesizing with unseen input data. In order to mitigate the limitation in robustness, we propose Mel Calibrator, a linear compression or expansion of the audio to the text's length. Only with the total lengths of both input text and audio, Mel Calibrator averages frames or repeats a frame of audio to assign them to a single phone by the ratio of the size of phoneme sequence over audio frame length. In this physically scaling method, the aligning process becomes a simple frame-wise bottleneck of incoming audio, requiring neither attention nor forced alignment. There are two advantages over the attention mechanism in terms of our objective: it does not bring any robustness issue against the non-autoregression, and it does not being exposed to the text so that audio-related style factors only come from the audio.

2.1.2. Information Bottleneck under Supervision

Encoders of prior unsupervised speech decomposition [19, 20] need a tight bottleneck both channel-wise and frame-wise to deliver the necessary information from the source audio. But it makes training difficult and time-consuming. Under the supervision, encoders can utilize pre-obtained features from incoming audio such as pitch contour and energy. Since forced feature selection is not needed, excessive regulation will lead to information shortage and performance degradation [19]. In our model, the bottleneck is adequately mitigated to fit the information flow relishing the supervision during style factor modeling.

2.2. Model Architecture

Figure 1 shows an overview architecture of the proposed model. STYLER models total six style factors: text, duration, pitch, speaker, energy, and noise. Text is regarded as a style factor equal to other audio related factors. To achieve rapidity, robustness, expressivity, and controllability at the same time, the elements of non-autoregressive TTS[3] [12, 17] and speech decomposition approaches[4] [19, 20] are adequately selected and transformed into STYLER.

2.2.1. Encoders

Text encoder has two 256 dimensional Feed-Forward Transformer (FFT) blocks [12] with four heads and takes phoneme sequence as input text. Since the text is encoded solely by the text encoder, the content of audio is removed through the bottleneck in other encoders. Duration, pitch, energy, and noise encoder have three 5x1 convolution layers followed by a group normalization [21] of size 16, and channel-wise bottleneck by two layers bidirectional-LSTM of size 64 except the duration encoder where the size is 80. The dimension of the convolution layer is 256 for duration and noise, 320 for pitch and energy. Mel Calibrator is applied after the convolution stack. Encoder outputs are upsampled channel-wise before being sent to each predictor by linear layer with ReLU activation. Length regulator [12, 17] is applied as a frame-wise upsampling to repeat frames for each phone counted by duration encoding.

The duration and noise encoder take mel-scaled spectrogram (mel-spectrogram). The pitch encoder takes a speaker normalized pitch contour ($mean = 0.5$, $std = 0.25$) so that speaker-independent pitch contour is modeled. Speaker encoding is transferred from pre-trained speaker embedding, as presented in [22]. In this work, Deep Speaker [23] is employed since it is faithful to embed speaker identities without additional features such as noise robustness. The energy encoder takes scaled energy from 0 to 1 to ease the model training. After extracted from audio as in [17], both pitch and energy input are quantized in 256 bins one-hot vector and then processed by encoders. Note that all inputs go through channel-wise bottleneck only. This is because the audio is already processed by the Mel Calibrator, and an additional frame-wise bottleneck acts as the excessive regulation discussed in Section 2.1.2.

2.2.2. Decoders

STYLER has three predictors [17], each for the duration, pitch, and energy. The predictors need to consume all necessary information since they predict real values rather than latent codes. The input contains the sum of the corresponding encoding with text encoding. The text encoding is projected to four-dimensional space to balance the dependency. The pitch predictor receives speaker encoding as an additional input to predict the final pitch contour. We empirically find that adding speaker encoding to both downsampled and upsampled pitch encoding improves the decomposition of speaker identity.

Decoder, which predicts the final mel-spectrogram from disentangled style factors, has four 256 dimensional FFT blocks with four multi heads. The decoder's input is a combination of text encoding, pitch embedding (embedding of pitch predictor output), energy embedding (embedding of energy predictor output), and speaker encoding. Speaker encoding is included be-

---

[3]https://github.com/ming024/FastSpeech2
[4]https://github.com/auspicious3000/SpeechSplit

cause information between input and target should be matched for the desired mapping.

2.3. Noise modeling

On the way of improving the noise robustness of STYLER, the noise can also be decomposed from audio without additional labels. In our model, noise is one of the remaining factors that have been explicitly encoded in model encoders, and it can be defined based on its residual property, excluding other style factors. In order to model noise by the definition, however, other encoders must be constrained to not include noise information even from the noisy input.

2.3.1. Domain Adversarial Training

Not including noise information means extracting noise-independent features. In the TTS domain, this has been tackled by applying Domain Adversarial Training (DAT) [24, 25]. Similar to previous works, the augmentation label and a gradient reversal layer (GRL) are introduced, which is jointly trainable in our model. The label of each predictor acts as a class label. As shown in Figure 1, DAT is applied to every audio-related encoding except noise encoder. After passing through GRL, each encoding is consumed by the augmentation classifier to predict the augmentation posterior (original/augmented). The augmentation classifier consists of two fully connected layers of 256 hidden size, followed by layer normalization [26] and ReLU activation. Note that each encoder except the noise encoder is now noise-independent, so each predictor's output is compared to clean labels rather than noisy ones.

2.3.2. Residual Decoding

According to the definition of noise and the fact that an explicit label is not given, a novel pipeline called Residual Decoding is designed. It contains two phases: clean decoding and noisy decoding. In clean decoding, all noise-independent encodings are taken to predict a clean mel-spectrogrom, and in noisy decoding, noise encoder output is added to noise-independent encodings to predict a mel-spectrogrom with noise. Since only the noise encoder has to be updated at noisy decoding, the gradient does not flow through the other encoders. Residual Decoding can be seen as implicit supervision where the noise is directly compelled to focus on the leftover part of the audio with no explicit label.

2.4. Loss Calculation

The total loss of the model without noise modeling is:

$$Loss_{clean} = l_{mel-clean} + l_{duration} + l_{pitch} + l_{energy} \quad (1)$$

where $l_{mel-clean}$ is mean square error between predicted mel-spectrogram and target, and predictor losses ($l_{duration}$, l_{pitch}, and l_{energy}) are calculated by mean absolute error. Note that each extraction method of duration, pitch, and energy and the loss is the same as [17], which means that the proposed model can disentangle style factors with no additional label. The loss, including noise modeling, then becomes as follows:

$$Loss_{total} = Loss_{clean} + l_{mel-noisy} + l_{aug} \quad (2)$$

where $Loss_{clean}$ refers to Equation 1, $l_{mel-noisy}$ is from noisy decoding calculated by the same way of $l_{mel-clean}$ in Equation 1, and l_{aug} is the sum of each augmentation classifier loss in Section 2.3.1.

3. Experiment

3.1. Experiment Setup

Throughout the experiment, we train and evaluate models on VCTK corpus [27]. For the noisy dataset, we augment the original dataset by mixing each utterance with a randomly selected piece of background noise from WHAM! dataset [28], at a random signal-to-noise ratio (SNR) in a range from 5 to 25. Raw audio is resampled at 22050Hz sampling rate and preprocessed to 1024 filter lengths, 1024 window sizes, and 256 hops lengths.

We divide the dataset into train, validation, and test set. In the first two sets, 44 speakers of each gender are randomly selected to balance speaker statistics. The remaining speakers are in the test set. The processed dataset is about 26 hours in total, containing 35805, 89, and 8345 utterances for the train, validation, and test set, respectively.

For the baseline, we use the following models to compare with STYLER in four aspects of our goal.

- Mellotron [29]: Tacotron2 GST [2, 7] based autoregressive model, with ability to condition pitch and duration explicitly. We use this baseline to compare speed and robustness of expressive and controllable TTS.

- FastSpeech2 [17]: TTS with non-autoregressive decoding and supervision in pitch, duration, energy control. We select this baseline to compare expressiveness and controllability of reading style non-autoregressive TTS.

STYLER is optimized by the same optimizer and scheduler in [17] with a batch size of 16. For the fair comparison, we apply pretrained ResCNN Softmax+Triplet model[5] for the speaker embedding identically to all models. We use pretrained UNIVERSAL_V1 model[6] of HiFi-GAN [4] as a vocoder.

3.2. Evaluation Metrics

We evaluate our model in four aspects: rapidity, robustness, expressiveness, and controllability. Training and inference time is strictly measured to prove the rapidity. Mean opinion score (MOS) is conducted to show the naturalness and robustness. We further investigate edge cases of weak robustness. Comparative MOS (CMOS) is conducted on a style transfer task to compare the expressiveness. For both MOS and CMOS, we use Amazon Mechanical Turk [30, 31]. Lastly, the controllability is demonstrated through an ablation study. There are two different speaker settings: seen speakers (S) and unseen speakers (US) in training time. And there are two different synthesis environments: parallel (P) indicates audio content and input text are identical, where nonparallel (NP) denotes the opposite.

3.3. Results

3.3.1. Rapidity and Naturalness

Table 1: *Time Measurement & MOS Results*

Model	Time (sec)		S		US	
	Training	Inference	MOS-P	MOS-NP	MOS-P	MOS-NP
FastSpeech2	0.102	0.019	3.753		3.750	
Mellotron	0.914	0.215	3.600	3.256	3.572	3.165
STYLER	**0.262**	**0.024**	**3.594**	**3.573**	**3.632**	**3.650**

Table 1 shows experiment results of model speed and audio naturalness. For the model speed, we measure average training

time per step during the first 1k steps and average inference time per single text from a set of speakers without vocoding. Even with increased times compared to FastSpeech2 due to the style factor modeling, STYLER shows faster training and inference speed about **3.49×** and **8.96×** than Mellotron.

Comparing MOS score against FastSpeech2 confirms that STYLER can synthesize natural speech even with increased model complexity. STYLER is ranked higher than Mellotron for all conditions except S & P where only a trivial gap exists. While both baselines show performance degradation under US or NP conditions, STYLER shows better naturalness for both cases. In our auxiliary experiment, STYLER begins to tilt on average style at a particular point of uncertainty, bringing more naturalness than expressive synthesis. It leads to higher MOS score in US than S with an increasing MOS score in US. In edge case investigation, Mellotron shows repeated and ignored words due to the broken alignment map, which is not the case in STYLER. The naturalness tendency and edge cases indicate that our model synthesizes more natural speech than Mellotron especially when unseen data is consumed as input, proving unseen data robustness.

3.3.2. Style Transfer

Table 2: *CMOS Results*

Model	S		US	
	CMOS-P	CMOS-NP	CMOS-P	CMOS-NP
STYLER	0	0	0	0
FastSpeech2	-0.303	-0.478	-0.388	-0.266
Mellotron	-0.334	-0.422	-0.25	-0.359

Table 2 reports CMOS of style transfer performance where STYLER score is fixed to 0. Comparison with FastSpeech2 and Mellotron denotes that our method represents better expressiveness than both baselines. Also, the style transfer performance decreases in all conditions of Mellotron, showing our model's robustness on unseen data over the baseline. In another auxiliary experiment, there exists a trade-off between the naturalness and style transfer performance according to the dependency of each model input. Naturalness score, in detail, is proportional to the influence of text rather than audio, while style transfer performance is inversely proportional. FastSpeech2 is an extreme case where the model has text dependency only, and hence the best naturalness yet in average style. The phenomenon is relatively severe in duration so that the duration modeling could be less precise than other style factors. Note that even with such a trade-off, STYLER outperforms Mellotron in all conditions.

3.3.3. Style Factor Modeling

Figure 2 shows our ablation study results as mel-spectrogram. Validating the functionality of style factor modeling, we synthesize speech while excluding each encoding one-by-one.

Synthesized from noisy reference audio (top-left), the resulting speech contains background noise (top-right) or not (middle-left) depending on whether the noise decoding is activated. Only the noise is synthesized when the noise encoding is solely activated (middle-right). From these results, it is proven that STYLER can produce clean audio from the noisy reference audio, confirming our model's noise-robustness. We also find that noise modeling predicts the different noise levels of input audio. Summing up, these experiments conclude that our novel noise modeling successfully decomposes noise from reference
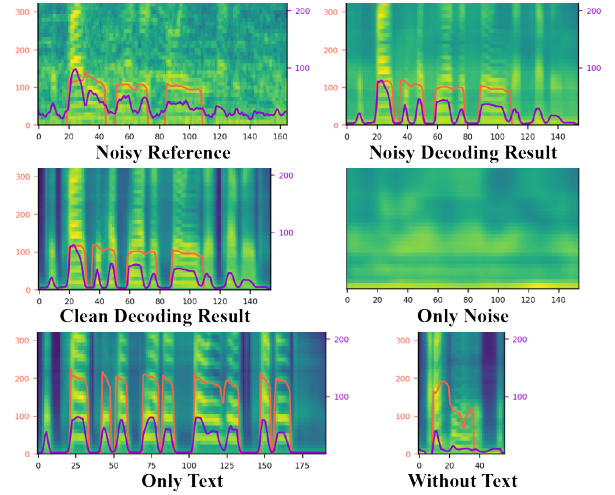

Figure 2: *Results of style factor decomposition. The orange line represents pitch contour, and the purple line represents energy. The content is "The party has never fully recovered." and spoken by a male speaker. Both input text and audio are not shown in the training time. The figure is best shown in color.*

audio like other style factors.

When the noise-independent style factors are ignored, the corresponding encodings become fixed style (bottom-left). Excluding pitch encoding, for instance, synthesizes speech has flat pitch contour. Also, given another speaker, the speaker's identity is changed while the other style factors are unchanged. When the text encoder is deactivated, audio-related style factors are modeled correctly, while synthesized speech is unintelligible (bottom-right). From all ablation study results, we conclude that encoders successfully capture expected style factors.

In FastSpeech2, the duration, pitch, and energy are already controllable. However, the values are predicted from the same text and can be controlled only after being synthesized. In contrast, STYLER distributes the dependency equally among text and style factors and controls them at the input levels. Therefore, with disentangled style factors, STYLER can control output speech by input variants in addition to a single text, where all inputs can be of all different sources.

4. Conclusion

In this paper, we propose STYLER, a non-autoregressive TTS framework with style factor modeling that achieves rapidity, robustness, expressivity, and controllability at the same time. It shows high performance and naturalness throughout various experiments. As discussed, however, our model has a trade-off between style transfer performance and speech naturalness. Our next plan is to resolve this trade-off, extending to different datasets and languages. We believe that STYLER can become a cornerstone for applications and researches in TTS community.

5. Acknowledgements

6. References

[1] Y. Wang, R. Skerry-Ryan, D. Stanton, Y. Wu, R. J. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio, Q. Le, Y. Agiomyrgiannakis, R. Clark, and R. A. Saurous, "Tacotron: Towards end-to-end speech synthesis," in *Proc. Interspeech 2017*, 2017, pp. 4006–4010.

[2] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. 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.

[3] R. Prenger, R. Valle, and B. Catanzaro, "Waveglow: A flow-based generative network for speech synthesis," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 3617–3621.

[4] J. Kong, J. Kim, and J. Bae, "HiFi-GAN: Generative adversarial networks for efficient and high fidelity speech synthesis," in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 17 022–17 033.

[5] Z. Hodari, O. Watts, and S. King, "Using generative modelling to produce varied intonation for speech synthesis," in *Proc. 10th ISCA Speech Synthesis Workshop*, 2019, pp. 239–244.

[6] D. Stanton, Y. Wang, and R. Skerry-Ryan, "Predicting expressive speaking style from text in end-to-end speech synthesis," in *2018 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2018, pp. 595–602.

[7] 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 *Proceedings of the 35th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 80. PMLR, 10–15 Jul 2018, pp. 5180–5189.

[8] Y. Zhang, S. Pan, L. He, and Z. 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)*, 2019, pp. 6945–6949.

[9] W.-N. Hsu, Y. Zhang, R. Weiss, H. Zen, Y. Wu, Y. Cao, and Y. Wang, "Hierarchical generative modeling for controllable speech synthesis," in *International Conference on Learning Representations*, 2019.

[10] Y. Lee and T. Kim, "Robust and fine-grained prosody control of end-to-end speech synthesis," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 5911–5915.

[11] G. Sun, Y. Zhang, R. J. Weiss, Y. Cao, H. Zen, and Y. 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.

[12] 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 *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[13] S. Bengio, O. Vinyals, N. Jaitly, and N. Shazeer, "Scheduled sampling for sequence prediction with recurrent neural networks," in *Advances in Neural Information Processing Systems*, vol. 28, 2015.

[14] M. Ranzato, S. Chopra, M. Auli, and W. Zaremba, "Sequence level training with recurrent neural networks," in *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016.

[15] F. Locatello, S. Bauer, M. Lucic, G. Raetsch, S. Gelly, B. Schölkopf, and O. Bachem, "Challenging common assumptions in the unsupervised learning of disentangled representations," in *international conference on machine learning*. PMLR, 2019, pp. 4114–4124.

[16] K. Peng, W. Ping, Z. Song, and K. Zhao, "Non-autoregressive neural text-to-speech," in *International Conference on Machine Learning*. PMLR, 2020, pp. 7586–7598.

[17] Y. Ren, C. Hu, X. Tan, T. Qin, S. Zhao, Z. Zhao, and T.-Y. Liu, "Fastspeech 2: Fast and high-quality end-to-end text to speech," in *International Conference on Learning Representations*, 2021.

[18] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[19] K. Qian, Y. Zhang, S. Chang, X. Yang, and M. Hasegawa-Johnson, "Autovc: Zero-shot voice style transfer with only autoencoder loss," in *International Conference on Machine Learning*. PMLR, 2019, pp. 5210–5219.

[20] K. Qian, Y. Zhang, S. Chang, M. Hasegawa-Johnson, and D. Cox, "Unsupervised speech decomposition via triple information bottleneck," in *International Conference on Machine Learning*. PMLR, 2020, pp. 7836–7846.

[21] Y. Wu and K. He, "Group normalization," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 3–19.

[22] Y. Jia, Y. Zhang, R. Weiss, Q. Wang, J. Shen, F. Ren, z. Chen, P. Nguyen, R. Pang, I. Lopez Moreno, and Y. Wu, "Transfer learning from speaker verification to multispeaker text-to-speech synthesis," in *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[23] C. Li, X. Ma, B. Jiang, X. Li, X. Zhang, X. Liu, Y. Cao, A. Kannan, and Z. Zhu, "Deep speaker: an end-to-end neural speaker embedding system," *arXiv preprint arXiv:1705.02304*, vol. 650, 2017.

[24] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, "Domain-adversarial training of neural networks," *The journal of machine learning research*, vol. 17, no. 1, pp. 2096–2030, 2016.

[25] W.-N. Hsu, Y. Zhang, R. J. Weiss, Y.-A. Chung, Y. Wang, Y. Wu, and J. Glass, "Disentangling correlated speaker and noise for speech synthesis via data augmentation and adversarial factorization," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 5901–5905.

[26] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016.

[27] J. Yamagishi, C. Veaux, K. MacDonald *et al.*, "Cstr vctk corpus: English multi-speaker corpus for cstr voice cloning toolkit (version 0.92)," *https://datashare.ed.ac.uk/handle/10283/3443*, 2019.

[28] G. Wichern, J. Antognini, M. Flynn, L. R. Zhu, E. McQuinn, D. Crow, E. Manilow, and J. Le Roux, "Wham!: Extending speech separation to noisy environments," in *Proc. Interspeech*, Sep. 2019.

[29] R. Valle, J. Li, R. Prenger, and B. 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.

[30] M. Chu and H. Peng, "Objective measure for estimating mean opinion score of synthesized speech," Apr. 4 2006, uS Patent 7,024,362.

[31] F. Ribeiro, D. Florêncio, C. Zhang, and M. Seltzer, "Crowdmos: An approach for crowdsourcing mean opinion score studies," in *2011 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2011, pp. 2416–2419.