INTERSPEECH 2021
30 August – 3 September, 2021, Brno, Czechia

Expressive Text-to-Speech using Style Tag

Minchan Kim[1], *Sung Jun Cheon*[1], *Byoung Jin Choi*[1], *Jong Jin Kim*[2], *Nam Soo Kim*[1]

[1]Department of Electrical and Computer Engineering and INMC,
Seoul National University, Seoul, South Korea
[2]SK Telecom, Seoul, South Korea

{mckim, sjcheon, bjchoi}@hi.snu.ac.kr, kimjj.geek@sk.com, nkim@snu.ac.kr

Abstract

As recent text-to-speech (TTS) systems have been rapidly improved in speech quality and generation speed, many researchers now focus on a more challenging issue: expressive TTS. To control speaking styles, existing expressive TTS models use categorical style index or reference speech as style input. In this work, we propose StyleTagging-TTS (ST-TTS), a novel expressive TTS model that utilizes a style tag written in natural language. Using a style-tagged TTS dataset and a pre-trained language model, we modeled the relationship between linguistic embedding and speaking style domain, which enables our model to work even with style tags unseen during training. As style tag is written in natural language, it can control speaking style in a more intuitive, interpretable, and scalable way compared with style index or reference speech. In addition, in terms of model architecture, we propose an efficient non-autoregressive (NAR) TTS architecture with single-stage training. The experimental result shows that ST-TTS outperforms the existing expressive TTS model, Tacotron2-GST in speech quality and expressiveness.

Index Terms: speech synthesis, expressive TTS, language model, non-autoregressive TTS

1. Introduction

Recently, end-to-end (e2e) speech synthesis systems have achieved significant improvement and now show almost human-like speech quality. Among e2e speech synthesis systems, autoregressive (AR) models such as Tacotron2 [1] and Transformer TTS [2] first showed state-of-the-art performance by exploiting the attention mechanism. However, as AR models suffer from slow generation speed and a lack of stability due to attention failure, non-autoregressive (NAR) models [3–8] have been proposed in recent years. By explicitly modeling the duration of each text unit and generating speech in parallel, NAR models can synthesize speech much faster than real-time with comparable speech quality with AR models. Such development enables e2e speech synthesis systems to be successfully adopted in many real-world applications.

Meanwhile, to imitate the non-phonetic expressions of human utterance, expressive TTS [9–14] has been studied in line with the development of the e2e TTS systems. In expressive TTS, the speaking style is modeled in a supervised or unsupervised manner and usually combined with an existing TTS architecture. In terms of style control, there are two approaches for expressive TTS. Several models exploit the categorical style labels, which indicate speaking styles such as emotions [9–11]. Trained with a style-labeled dataset in a supervised manner, these models can control style using explicit labels. However, there are limitations in the diversity of expression, as these models can only express a few pre-defined styles. On the other hand, some models use reference speech as style input [12–14]. In these models, style information is extracted from the reference speech using a reference encoder and transferred to the generated speech. Although these methods don't require a labeled dataset and can express unbounded speaking styles, the extracted style information is not intuitive and interpretable. Moreover, choosing a reference speech every time is time-consuming and memory inefficient, which makes these methods less practical.

In this paper, we introduce a novel style interface for expressive TTS: style tag. Style tag is a short phrase or word representing the style of an utterance, such as emotion, intention, and tone of voice. As style is tagged in the natural language, it is intuitive and interpretable for controlling the speaking style. In addition, we propose Style Tagging TTS (ST-TTS), which is a novel non-autoregressive expressive TTS model utilizing style tag as a style input. Using a pre-trained language model as an interpreter for transforming natural language to linguistic embedding, ST-TTS can learn the relationship between linguistic embedding and style embedding space. Due to the generalization capabilities of the language model, even a style tag unseen in the training dataset can be expressed by ST-TTS. Besides, ST-TTS also has several advantages in model architecture. To find alignments between text and speech, ST-TTS uses Monotonic Alignment Search (MAS) algorithm [5] and normalizing flow (NF) [15, 16] as an aligner. As the aligner is jointly optimized with the entire TTS system, ST-TTS can be trained in single-stage, unlike other NAR models such as FastSpeech [3, 4] and SpeedySpeech [8]. Additionally, at the inference phase, we can optionally use either style tag or reference speech as a style input by building a shared embedding space of reference speech and style tag.

This paper is organized as follows. In Section 2, we first describe the dataset we used. Then we explain the proposed ST-TTS model in section 3, and experiment results are reported in section 4. Lastly, the conclusion and discussion are covered in section 5.

2. Style Tagging Dataset

As existing expressive TTS datasets [17–19] have only a few numbers of speaking styles, a dataset with much more various styles is required for our proposed method. We introduce a Korean Style Tagging TTS dataset: FSNR0[1]. FSNR0 consists of {speech, transcript, style tag} tuples and the entire transcripts and style tags are written in Korean. The transcripts are recorded by a voice actress expressing given style tags.

[1]FSNR0 is a part of SMART-TTS corpus, a Korean TTS corpus for an expressive TTS project named SMART-TTS. We are planning to release this corpus as an open-resource.

Copyright © 2021 ISCA

http://dx.doi.org/10.21437/Interspeech.2021-465

In FSNR0, There are 327 style tags including various emotions, intentions, voice tone and speed. Each style tag appears from one to hundreds of times in the whole dataset except for the normal reading book style, which takes up about 25% of the total data. The entire speech data is about 26 hours long and consists of about 18,700 sentences. Several examples of style tags are shown in Table 1. In training, the style tags are augmented by several rules such as switching adverbs and adjectives, and extending keywords to plausible phrase forms. One of the augmented style tag is sampled every iteration.

Table 1: *Examples of style tag. We additionally provide the translated ones to help understanding.*

style tag	translated	style tag	translated
다정하게	with affection	매정하게	heartless
화가난 듯	seem angry	기쁜 듯	pleased
씁쓸한 듯	bitter	다급하게	in a hurry
큰 소리로	in a loud voice	속삭이듯	whispering
졸린 듯	sleepy	술취한 듯	drunken

3. Style Tagging TTS

3.1. Model architecture

In this section, we introduce the ST-TTS architecture. Our proposed model takes grapheme sequence and style tag as input and returns log mel-spectrogram. Reference speech is additionally used for training and optionally used for inference. The entire structure is based on non-autoregressive feedforward TTS models [3, 4, 8], and end-to-end trainable by jointly optimizing duration information with the entire system. The entire architecture is shown in Figure 1, and described as follows.

3.1.1. Sentence BERT

As style tag is an infinite set, we used a pre-trained language model to embed a style tag to a meaningful space without Out-Of-Vocabulary (OOV) problem. It is the core component in ST-TTS that enables generalization of unseen style tags. We employed Sentence BERT (SBERT) [20] for our proposed model. BERT [21] is one of the most widely used representation learning for natural language process (NLP). It is composed of Transformer [22] blocks and trained with masking and slot filling task. Although BERT performs well in various NLP tasks such as machine translation, it lacks the ability to extract a sentence-level embedding. SBERT is a fine-tuned model of BERT optimized for sentence-level similarity using contrastive learning. SBERT shows better performance in measuring sentence similarity and clustering sentences than original BERT. We regard style tags as extremely short sentences and used pre-trained SBERT as a component for the style encoder.

3.1.2. Style encoder

Style encoder is a module that extracts style embedding for the TTS system, and it consists of a reference encoder and a style tag encoder. Reference encoder extracts style embedding from reference speech. It has almost the same architecture with reference encoder of Global Style Tokens (GSTs) [12]. The difference is that the proposed module does not use style token, and batch normalization [23] layers are replaced with weight normalization [24]. Style tag encoder is a module for extracting a style embedding from a style tag. It consists of pre-trained SBERT and adaptation layers. As mentioned in 3.1.1, SBERT transforms style tag to semantic embedding. Adaptation layers are made up of three linear layers with ReLU activation and learn to map from linguistic semantic space to style embedding space.

In training, style embedding from reference encoder is used for TTS system, and then the embedding from style tag encoder learns reference embedding by mean squared error (MSE) loss named style embedding loss. This bi-modal conditioning method has the following advantages. 1) By building a bi-modal embedding space of style tag and reference speech, either style tag or reference speech can be used to extract a style embedding for inference. 2) Style tag embedding functions as an anchor of reference speeches with the same style tag, which means the reference encoder learns to form clusters of reference speeches with the same style tags. As speeches with similar speaking styles are mapped adjacently in embedding space by the reference encoder, style tags with similar meanings are also located closely in embedding space. This property helps to form a well-defined embedding space, and adaptation layers can easily learn the relationship between style domain and linguistic domain. While training, we freeze SBERT, and the other components are optimized for the entire TTS loss and the style embedding loss.

3.1.3. Text encoder

Text encoder takes a grapheme sequence and returns a text embedding sequence. It consists of residual dilated convolution blocks used in Parallel WaveGan[2] [25]. We used 12 residual blocks with dilation rates (1, 2, 4) 4 times. The kernel size and hidden dimensions are 5 and 256 respectively.

The output embedding sequence is used for the aligner, duration predictor, and mel decoder. When used for mel decoder, the text embeddings are copied multiple times to match text and speech length using duration information from aligner or duration predictor.

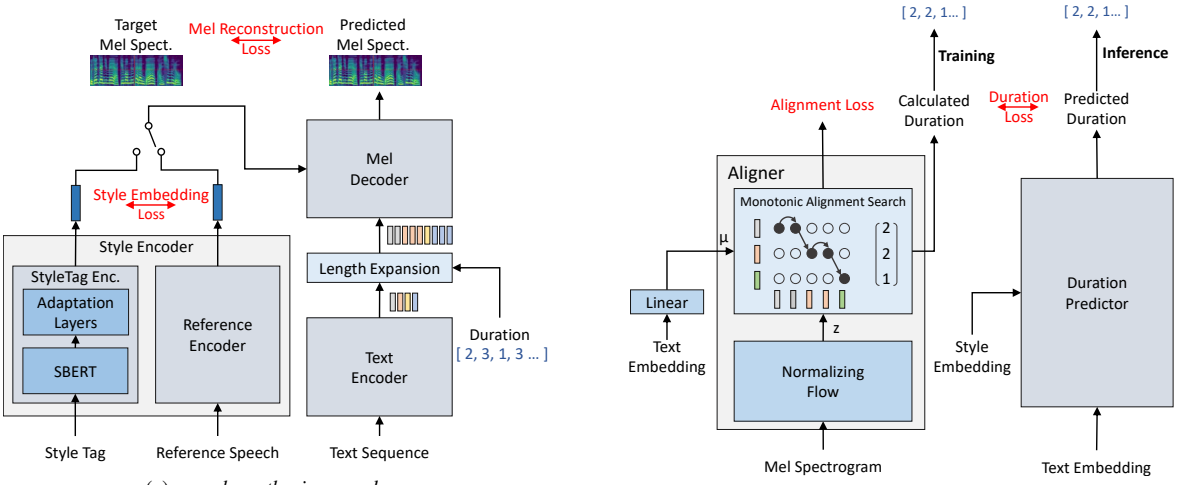
3.1.4. Aligner

Aligner calculates alignment between text and log mel-spectrogram and returns duration of each grapheme. For aligner, we exploit the method devised in Glow-TTS which uses normalizing flow (NF) and Monotonic Alignment Search (MAS) algorithm [5]. To calculate alignment, mel-spectrogram $x_{1:T}$ is first mapped to latent variable $z_{1:T}$ by NF $f_{NF}^{-1} : x \rightarrow z$. Text encoder output is then projected to the latent space with a linear layer and regarded as parameter $\mu_{1:N}$ of the latent variable. T and N denote the length of mel-spectrogram and text tokens respectively. For a possible alignment A, $A(j) = i$ represents $z_j \sim \mathcal{N}(\mu_i, \sigma_i)$, where σ_i of the entire frame is set to 1 for stable training. By change of variables formula used in NF, the conditional log likelihood of mel-spectrogram given text input $c_{1:N}$ and an alignment A can be expressed as Eq. (1) and Eq. (2):

$$\log P_X(x|c;A) = \log P_Z(z|c;A) + \log \left| \det \frac{\partial f_{NF}^{-1}(x)}{\partial x} \right|, \quad (1)$$

$$\log P_Z(z|c;A) = \sum_{j=1}^{T} \log \mathcal{N}(z_j; \mu_{A(j)}, \sigma_{A(j)}). \quad (2)$$

[2]referred code : https://github.com/kan-bayashi/ParallelWaveGAN



(a) *speech synthesis procedure*

(b) *modules for duration extraction and prediction*

Figure 1: *The entire architecture of ST-TTS*

The optimal alignment path A^* for maximizing Eq. (1) can be obtained by the MAS algorithm. In training, NF and text encoder are optimized for maximizing $logP_X(x|c; A^*)$, and duration for length expansion is calculated by summation A^* over mel-spectrogram axis.

In Glow-TTS, NF is itself a generator of mel-spectrogram, but we only exploit NF as an aligner whose role is just providing an accurate alignment. We used the same architecture with Glow-TTS[3] in a much smaller size. The NF is consists of 6 flow blocks, and the affine coupling layers in each flow block are a stack of 4 residual convolution layers with kernel size 5 and hidden dimension 128. Unlike Glow-TTS, we did not squeeze mel-spectrogram for better alignment.

As this aligning method only considers monotonic alignments between text and speech, ST-TTS can be trained much faster and robustly compared with the attention mechanism generally used for TTS alignment. In addition, it does not require auxiliary methods such as guided attention loss [26] or positional encoding [22] for attention mechanism.

3.1.5. Duration predictor

Duration predictor predicts the number of frames for each grapheme in log scale. Duration predictor is made up of the same residual convolution blocks used in the text encoder. As different style yields different speaking speed, duration predictor also takes a style embedding to predict style dependent duration. Style embedding is duplicated to the equal length of text embedding and conditioned to each residual block by the same method used in [25]. Duration predictor is composed of five residual blocks without dilation, and kernel size and hidden dimension are 5 and 256.

3.1.6. Mel decoder

Mel decoder transforms duration-expanded text embedding to log mel-spectrogram conditioned on a style embedding. It has the same architecture with the duration predictor in a much larger size. Mel decoder consists of 30 residual blocks with dilation rate (1, 2, 4, 8, 16) 6 times. The kernel size used in the mel decoder is 3, and the hidden dimension is 256.

[3]referred code : https://github.com/jaywalnut310/glow-tts

3.2. Training and inference

In training, the total loss is the sum of the following four losses.

- Mel reconstruction loss: mean-absolute-error (MAE) between predicted and target log mel-spectrogram.
- Duration loss: huber loss [27] between output of duration predictor and log duration acquired by aligner.
- Alignment loss: negative log likelihood loss for training aligner described in 3.1.3
- Style embedding loss : MSE loss between style embeddings from reference encoder and style tag encoder as described in 3.1.2

These losses are summed with equal weights and jointly optimized for the entire model except for SBERT in style encoder.

During inference, ST-TTS uses duration from the duration predictor. As the duration predictor predicts duration in log scale, it is transformed by exponential operation and rounded to the nearest integer. To prevent skipping text units, the predicted durations less than one are clamped to one. We can optionally use style embedding extracted from reference speech or style tag depend on user convenience, as mentioned in 3.1.2.

4. Experiments

4.1. Experimental setup

For our experiments, we used the FSNR0 dataset described in section 2. The audio is downsampled from 48 kHz to 22.05 kHz, and 80-dimensional log mel-spectrogram was used for the target and reference speech. For input text, we used Korean grapheme without G2P. Style tags are tokenized by pre-trained SBERT. We trained our model using the Adam optimizier [28] with learning rate of 0.02 and the Noam learning scheduler [22]. Total training took about 87 hours for 500k iteration with a single GeForce GTX 2080 GPU. As a vocoder, we used HiFiGAN [29] which shows state-of-the-art audio quality for speech synthesis. For the subjective evaluations, 18 native Korean participants joined the tests described in 4.2.1 and 4.2.2. The audio samples of ST-TTS are available on the demo website[4].

[4]demo site: https://gannnn123.github.io/styletaggingtts-demo

4.2. Evaluation

4.2.1. Audio quality

We measure the 5 scale Mean Opinion Score (MOS) to estimate the speech quality of the proposed model. Since ST-TTS is the first TTS model which can control style with the undefined label or reference speech, we chose Tacotron2-GST [12] as a baseline. The result of five groups: ground truth (GT), GT with vocoder reconstruction, Tacotron2-GST, ST-TTS (reference speech), ST-TTS (style tag) is shown in Table 2. ST-TTS (reference speech) and ST-TTS (style tag) mean ST-TTS using reference speech and style tag for style input respectively. The reference speech used for Tacotron2-GST and ST-TTS (reference speech) is the target speech itself because it shows better performance for these models. We tested 30 samples each for normal reading book style samples and expressive samples with style tags, randomly chosen from the test set. According to Table 2, ST-TTS with either style tag or reference speech shows better speech quality than the baseline.

Table 2: *Mean Opinion Scores with 95% confidence intervals*

Method	Normal	Expressive
GT	4.77 ± 0.04	4.80 ± 0.04
GT (+ HifiGAN)	4.64 ± 0.05	4.67 ± 0.05
Tacotron2-GST	3.37 ± 0.08	3.44 ± 0.08
ST-TTS (reference)	4.15 ± 0.06	4.12 ± 0.06
ST-TTS (style tag)	4.17 ± 0.06	4.08 ± 0.06

4.2.2. Preference test

To demonstrate that the style tag can control the style of generated speech intuitively, we tested a preference test with ST-TTS and Tacotron2-GST. In this test, the participants listened to two samples from ST-TTS and Tacotron2-GST, and rated which sample reflected the given style tag better in 7 scale[5]. As Tacotron2-GST cannot directly use the style tag, we used the average style token weights of 10 utterances tagged with the given style tag. We regarded these weights as a representation of the given style tag for Tacotron2-GST. We tested 30 sentences with randomly selected style tags from the test set. The transcripts are sampled from another dataset. As a result, ST-TTS got the preference score 1.37 ($\pm$0.14, 95% confidence interval), where the score is ranged in (-3, 3), and a higher score means better performance. This result shows that ST-TTS can express various speaking styles recognized as reflecting style tags.

4.2.3. Visualization of style tag embedding

In Figure 2, we visualized the style embeddings of style tags using t-SNE [30]. There are all the style tags in the dataset and several style tags unseen in training. This visualization shows that style tags with similar meanings are embedded adjacently. Particularly, we can see that even the unseen style tags are properly located in the embedding space, which means the available style tag is not limited to the training dataset.

[5]The score is rated in the criterion: A is ( ) than B. Each score is described below.
-3: much worse, -2: worse, -1: little worse, 0: about same, 1: little better, 2: better, 3: much better

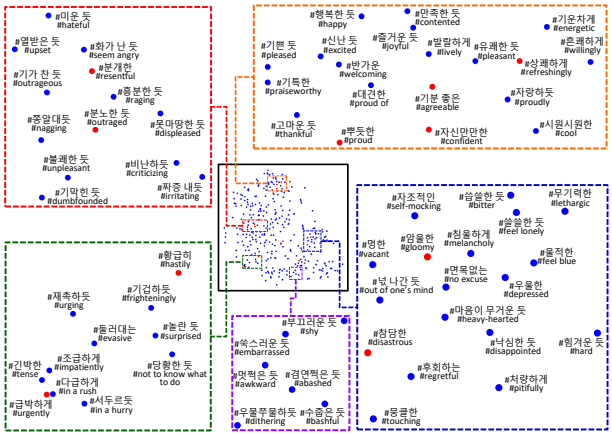


Figure 2: *Style embedding visualization using t-SNE. The entire t-SNE plot is at the center, and five local regions are enlarged for closer observation. The blue points and red points indicate the style tags seen and unseen in training, respectively. The translations with similar nuance are provided to help understanding.*

5. Conclusion and Discussion

In this paper, we proposed a novel expressive TTS model named ST-TTS, which uses style tag as a style interface. ST-TTS has several advantages in convenience and performance. 1) As style tag is written in natural language, users can intuitively control the style of generated speech with ST-TTS. The language model, SBERT enables ST-TTS to express even the style tags unseen during training. 2) ST-TTS has a bi-modal embedding space of reference speech and style tag, so either reference style tag or reference speech can be used in inference. 3) ST-TTS is a non-autoregressive TTS model which can be trained in single-stage. For this reason, ST-TTS can be easily trained and has fast generation speed.

As ST-TTS is the first TTS model that uses style tag, there are still remaining research topics as our future works. The most important issue in ST-TTS is the generalization performance for style tags. There are countless style tags in natural language, and the model should express as many style tags as possible for human-level expressiveness. This issue can be addressed in two directions. In terms of data configuration, it is important to use as many style tags as possible in training. We can augment style tags using synonyms to expand the covered linguistic embedding region. Next, as style tags are usually much shorter than common sentences, investigating a language model which can embed short phrases better can be a solution for better generalization. Meanwhile, there may be styles that cannot be expressed with a single style tag. We are planning to consider the methods using multi-style tags to express various characteristics together.

6. Acknowledgements

This work was supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) funded by the Korea Government (MSIT) under Grant 2020-0-00059 (Deep learning multi-speaker prosody and emotion cloning technology based on a high quality end-to-end model using small amount of data).

7. References

[1] 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.

[2] N. Li, S. Liu, Y. Liu, S. Zhao, and M. Liu, "Neural speech synthesis with transformer network," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 6706–6713.

[3] Y. Ren, Y. Ruan, X. Tan, T. Qin, S. Zhao, Z. Zhao, and T.-Y. Liu, "Fastspeech: Fast, robust and controllable text to speech," *arXiv preprint arXiv:1905.09263*, 2019.

[4] Y. Ren, C. Hu, T. Qin, S. Zhao, Z. Zhao, and T.-Y. Liu, "Fastspeech 2: Fast and high-quality end-to-end text-to-speech," *arXiv preprint arXiv:2006.04558*, 2020.

[5] J. Kim, S. Kim, J. Kong, and S. Yoon, "Glow-tts: A generative flow for text-to-speech via monotonic alignment search," *arXiv preprint arXiv:2005.11129*, 2020.

[6] C. Miao, S. Liang, M. Chen, J. Ma, S. Wang, and J. Xiao, "Flow-tts: A non-autoregressive network for text to speech based on flow," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 7209–7213.

[7] J. Donahue, S. Dieleman, M. Bińkowski, E. Elsen, and K. Simonyan, "End-to-end adversarial text-to-speech," *arXiv preprint arXiv:2006.03575*, 2020.

[8] J. Vainer and O. Dušek, "Speedyspeech: Efficient neural speech synthesis," *arXiv preprint arXiv:2008.03802*, 2020.

[9] Y. Lee, A. Rabiee, and S.-Y. Lee, "Emotional end-to-end neural speech synthesizer," *arXiv preprint arXiv:1711.05447*, 2017.

[10] N. Tits, K. El Haddad, and T. Dutoit, "Exploring transfer learning for low resource emotional tts," in *Proceedings of SAI Intelligent Systems Conference*. Springer, 2019, pp. 52–60.

[11] N. Tits, F. Wang, K. E. Haddad, V. Pagel, and T. Dutoit, "Visualization and interpretation of latent spaces for controlling expressive speech synthesis through audio analysis," *arXiv preprint arXiv:1903.11570*, 2019.

[12] 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 *International Conference on Machine Learning*. PMLR, 2018, pp. 5180–5189.

[13] 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*. PMLR, 2018, pp. 4693–4702.

[14] Y. Jia, Y. Zhang, R. J. Weiss, Q. Wang, J. Shen, F. Ren, Z. Chen, P. Nguyen, R. Pang, I. L. Moreno *et al.*, "Transfer learning from speaker verification to multispeaker text-to-speech synthesis," *arXiv preprint arXiv:1806.04558*, 2018.

[15] L. Dinh, J. Sohl-Dickstein, and S. Bengio, "Density estimation using real nvp," *arXiv preprint arXiv:1605.08803*, 2016.

[16] D. P. Kingma and P. Dhariwal, "Glow: Generative flow with invertible 1x1 convolutions," *arXiv preprint arXiv:1807.03039*, 2018.

[17] A. Adigwe, N. Tits, K. E. Haddad, S. Ostadabbas, and T. Dutoit, "The emotional voices database: Towards controlling the emotion dimension in voice generation systems," *arXiv preprint arXiv:1806.09514*, 2018.

[18] S. R. Livingstone and F. A. Russo, "The ryerson audio-visual database of emotional speech and song (ravdess): A dynamic, multimodal set of facial and vocal expressions in north american english," *PloS one*, vol. 13, no. 5, p. e0196391, 2018.

[19] K. Zhou, B. Sisman, R. Liu, and H. Li, "Seen and unseen emotional style transfer for voice conversion with a new emotional speech dataset," *arXiv preprint arXiv:2010.14794*, 2020.

[20] N. Reimers and I. Gurevych, "Sentence-bert: Sentence embeddings using siamese bert-networks," *arXiv preprint arXiv:1908.10084*, 2019.

[21] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[22] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *arXiv preprint arXiv:1706.03762*, 2017.

[23] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International conference on machine learning*. PMLR, 2015, pp. 448–456.

[24] T. Salimans and D. P. Kingma, "Weight normalization: A simple reparameterization to accelerate training of deep neural networks," *arXiv preprint arXiv:1602.07868*, 2016.

[25] R. Yamamoto, E. Song, and J.-M. Kim, "Parallel wavegan: A fast waveform generation model based on generative adversarial networks with multi-resolution spectrogram," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6199–6203.

[26] H. Tachibana, K. Uenoyama, and S. Aihara, "Efficiently trainable text-to-speech system based on deep convolutional networks with guided attention," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4784–4788.

[27] P. J. Huber, "Robust estimation of a location parameter," in *Breakthroughs in statistics*. Springer, 1992, pp. 492–518.

[28] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[29] J. Kong, J. Kim, and J. Bae, "Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis," *arXiv preprint arXiv:2010.05646*, 2020.

[30] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne." *Journal of machine learning research*, vol. 9, no. 11, 2008.