IMPROVING UNSUPERVISED STYLE TRANSFER IN END-TO-END SPEECH SYNTHESIS WITH END-TO-END SPEECH RECOGNITION

Da-Rong Liu, Chi-Yu Yang, Szu-Lin Wu, Hung-Yi Lee

National Taiwan University

{givebirthday, yangchiyi10, riviera1020, tlkagkb93901106}@gmail.com

ABSTRACT

End-to-end TTS model can directly take an utterance as reference, and generate speech from the text with prosody and speaker characteristics similar to the reference utterance. Ideally, the transcription of reference utterance does not need to match the text to be synthesized, so unsupervised style transfer can be achieved. However, in the previous model, because only the matched text and speech are used in training, given unmatched text and speech during testing would make the model synthesize blurry speech. In this paper, we propose to mitigate the problem by using the unmatched text and speech during training, and using the ASR accuracy of an end-to-end ASR model to guide the training procedure. The experimental results show that with the guidance of end-to-end ASR, both the ASR accuracy (objective evaluation) and the listener preference (subjective evaluation) of the speech generated by TTS model are improved. Moreover, we propose attention consistency loss as regularization, which is shown to accelerate the training.

Index Terms— Text-to-Speech, Automatic Speech Recognition

1. INTRODUCTION

End-to-end text-to-speech (TTS) has made great progress in recent years. The original end-to-end TTS models only train and generate voice for single speaker [1, 2, 3, 4]. Then several modifications were proposed to generate speech conditioned on different speakers [1, 2, 5, 6, 7]. However, these models often require training data with speaker labels, and cannot generate the voice of the speakers not in the training set. The end-to-end TTS can be more flexible if the model can directly take an utterance as reference, and generate speech from the text with prosody and speaker characteristics similar to the reference utterance. In this way, unsupervised voice style transfer can be achieved [4]. Once a speaker's utterance is provided as the reference utterance, TTS model can generate the voice of the speaker even though his or her voice is not in the training data. The above scenario is shown in the upper part of Fig. 1.

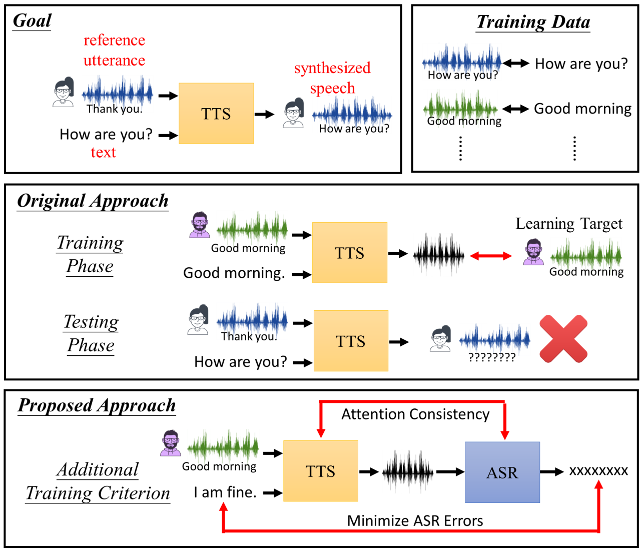


Fig. 1: The framework of the proposed approach.

Reference-encoder-based TTS models [8, 9] are proposed to achieve the goal. These models are trained from a set of utterances and their corresponding text transcriptions as in the upper-right corner of Fig. 1. As shown in the middle of Fig. 1, in training, the model takes a pair of utterance and text from training data as input. The input utterance is considered as the reference utterance. The model directly uses an encoder network to encode the prosody and speaker information of the reference utterance, and obtains reference embedding. The model learns to reconstruct the input utterance based on the reference embedding and the input text. During the testing phase, given a reference utterance and text (the text is not the transcription of the reference utterance), hopefully model can synthesize the speech of the given text using the characteristics of the reference utterance.

During the training phase, the reference utterance and the input text are paired, that is, the text is the transcription of the reference utterance. However, the input utterance and text are no longer paired when testing. Because the unpaired inputs are never seen during training, the generated voice will be blurry. Moreover, without carefully designing the network architecture of the reference encoder, the reference encoder may store some text information rather than prosody and speaker information in the reference embedding to reconstruct the input utterance. Therefore, the generated voice will even be influenced by the content of reference utterance. Carefully controlling the capacity of the reference encoder and selecting the parameters can mitigate the above issue to some extent [8, 9].

In this paper, we propose a more effective way to address the issue as shown in the lower part of Fig. 1. During training, we can sample unpaired utterance and text as the input of the TTS model, but there is no learning target. Therefore, we use an additional Automatic Speech Recognition (ASR) loss as learning target. We first train an end-to-end attention-based seq2seq ASR model. In TTS training, when the reference utterance and the text are unpaired, we feed the generated speech into the ASR model, and the TTS model learns to make the speech recognized as the input text. This additional loss can prevent the reference encoder from encoding any text information because if it does so, the ASR accuracy will be low with unpaired input. Both TTS and ASR models are end-to-end models, so the TTS model can be trained via back-propagation. Besides, we propose a novel regularization – attention consistency loss. Both the ASR and the TTS are attention-based seq2seq models. When the TTS model attends on a certain input character and generates an acoustic feature frame, the ASR model should attend on this frame when predicting the same character.[10, 11] Therefore, we constrain the attention weights of the ASR and TTS models to approximately fulfill the above assumption. This approach accelerates the convergence during training.

2. RELATED WORK

Most multi-speaker TTS models [1, 2, 6] require speaker labels to learn a speaker embedding matrix. Recently, some work try to improve multi-speaker TTS with speaker verification [12, 13]. A speaker verification model [14, 15] can be pretrained to extract the speaker characteristics from the reference utterance [16]. However, training a speaker verification model needs additional speaker labels. Cluster-based method [17, 18] uses some features, such as i-vector [19, 20], to cluster the training data, and trains a TTS model for each partition, but the clustering procedure and TTS are considered separately. Several works attempt to model the prosody or speaker information and speech synthesis simultaneously. The reference encoder is introduced which has been proved that it can be used to transfer prosody from a reference utterance, but speaker labels are still required [8]. GST-tacotron [9] further extends the previous work and adds a style token layer to disentangle the style into several embeddings. However, we found that these reference-encoder-based models usually generate blurry speech missing some characters in the text. To successfully train the models, one needs very carefully parameter tuning, dataset selection and trick usage. To mitigate the problem above, we extend GST-tacotron by adding an ASR. By making the generated speech recognized well by the ASR, we can make the generated speech clear and complete.

Before this work, several works have been proposed to improve the parametric speech synthesis via the additional discriminator. Anti-spoofing Verification (ASV) [21] is used as a discriminator to provide an additional constraint during training to make the vocoder parameters more realistic [22]. Generative adversarial network (GAN) [23] is used to generate glottal waveform, which can solve the problem that the model can only produce average waveforms when using only squared error loss [24, 25]. However, as far as we know, no previous works try to add discriminator on end-to-end TTS model.

3. END-TO-END TTS MODEL

We use GST-tacotron [8] as our reference-encoder-based TTS model. Given a character sequence x and the mel spectrogram of the reference utterance y_{ref}, GST-tacotron predicts the mel spectrogram y and linear spectrogram z. The training data for GST-tacotron is represented as $\{x^i, y^i, z^i\}_{i=1}^N$, where x^i is the manual transcription of the i-th utterance in the training set, while y^i and z^i are the mel spectrogram and linear spectrogram of the i-th utterance respectively. To train GST-tacotron, the model takes x^i and y^i as input and learns to reconstruct y^i and z^i. y^i is used as the input reference utterance y_{ref} and output target at the same time during training. Figure 2 shows the network architecture of GST-tacotron.

3.1. Reference encoder

The reference encoder, which is illustrated in Figure 2 (a) [26], takes the reference mel spectrogram y_{ref} as input, and compresses it into a fixed-length vector, called reference embedding. The reference mel spectrogram is first passed into a convolutional stack and an RNN, interleaved with batch normalization, to get a fixed-length vector. Then the fixed-length vector is passed through a style token layer [9] to get the final reference embedding. The style token layer is made up of a bank of style token embeddings and an attention module. The fixed-length vector is used as the key vector to compute the attention weight for each style token embedding via the attention module. The output is the weighted sum of the style token embeddings.

3.2. GST-Tacotron

The architecture and hyperparameters of the model in Figure 2 (b-1) is the same as previous work [4]. The text encoder first encodes the character sequences into sequential representations. Then these representations are concatenated with the reference embedding. An attention-based decoder is used to

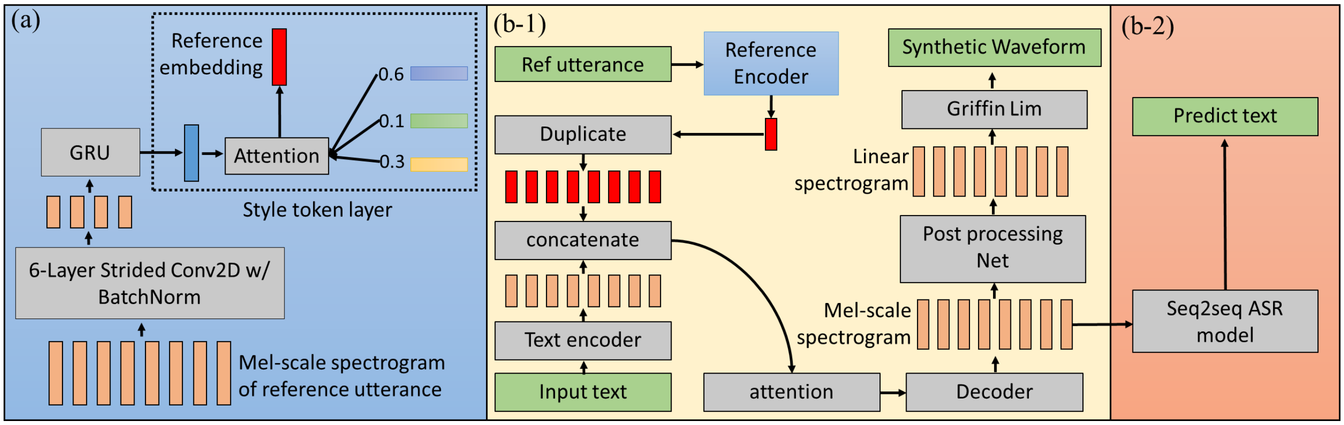


Fig. 2: (a) The architecture of the reference encoder. (b) The whole framework composed of two models: TTS model in (b-1) and ASR model in (b-2).

generate the mel spectrogram $\hat{y}$. Post processing net, which is a CBHG module [4], generates the linear spectrogram $\hat{z}$ from $\hat{y}$. The training loss is defined as:

$$l_{tts} = L_{mse}(y, \hat{y}) + L_{mse}(z, \hat{z}) \tag{1}$$

where $\hat{y}$ and $\hat{z}$ are generated spectrograms, and y and z are learning targets. L_{mse} is the mean squared error function. Finally, we use the Griffin-Lim [27] algorithm to synthesize waveform from the predicted linear spectrogram $\hat{z}$.

4. PROPOSED APPROACHES

The framework is composed of two components, illustrated in Figure 2 (b): (1) A reference-encoder-based end-to-end TTS model, and (2) an end-to-end ASR model, which predicts character sequence from mel spectrogram. The TTS model we used here is GST-tacotron [8], which is already introduced in Section 3, while ASR model can be any end-to-end seq2seq model[28, 29, 30, 31, 32, 33, 34]. In this paper, we adopt Listen, Attend and Spell (LAS)[31], an attention-based seq2seq model. We first pretrain TTS and ASR model separately, then fine tune the TTS model with the guidance of ASR model. Following sections will first go over the algorithm, then we will go to details about the proposed attention consistency loss and sampling process.

4.1. Algorithm

The algorithm is given in Algorithm 1. In the training of the original GST-tacotron, the input text x and reference mel spectrogram y_{ref} are always paired because there is no training target for the unpaired cases. Here we propose to use a pretrained ASR model to provide training target for unpaired cases. For any character sequence x and reference mel spectrogram y_{ref} that are not paired, we want the synthesized speech to be recognized well by the ASR model. We define the ASR loss to be:

Algorithm 1 ASR guided tacotron.

```
Require: Training data $\{(x^i, y^i, z^i)\}_{i=1}^N$, where $x^i$ denotes
    character sequence, $y^i$ denotes mel spectrogram, and $z^i$
    denotes linear spectrogram.
pretrain ASR model $M_{asr}$ with $\{(x^i, y^i, z^i)\}_{i=1}^N$
pretrain TTS model $M_{tts}$ with $\{(x^i, y^i)\}_{i=1}^N$
for t = 0,...,$num\_iter$ do
    $x^i, y^i, z^i \leftarrow$ Sample a tuple from training data
    $\hat{y}, \hat{z} \leftarrow M_{tts}(x^i, y^i)$ // generate speech in paired case
    $l_{tts} = L_{mse}(y^i, \hat{y}) + L_{mse}(z^i, \hat{z})$
    Update $M_{tts}$ to minimize $l_{tts}$
    $x^j \leftarrow$ Randomly sample another character sequence
    $\hat{y}' \leftarrow M_{tts}(x^j, y^i)$ // generate speech in unpaired case
    $l_{asr} = L_{asr}(M_{asr}(\hat{y}'), x^j)$
    Update $M_{tts}$ to minimize $l_{asr}$
end for
```

$$\hat{y} = M_{tts}(x, y_{ref}), \tag{2}$$

$$l_{asr} = L_{asr}(M_{asr}(\hat{y}), x), \tag{3}$$

where M_{tts} and M_{asr} denote the TTS and ASR model respectively; $\hat{y}$ denotes the mel spectrogram of the generated speech; L_{asr} is the cross entropy loss function. The parameters of the TTS model are updated to minimize l_{asr}.

4.2. Attention Consistency

For an attention-based seq2seq model, in time step t, the decoder will generate an attention weight vector w_t. The dimension of the attention weight vector w_t is equivalent to the number of input tokens. For example, for a TTS model, the dimension of w_t is the number of input characters in the input text. We can define the attention matrix W as the concatenation of the attention weight vectors of all time steps:

$$W = [w_1, w_2, ..., w_T] \tag{4}$$

where T denotes the decoder sequence length. Here both the TTS model and the ASR model are attention-based seq2seq models. Let W_{tts} and W_{asr} represent the attention matrices of the TTS model and ASR model respectively.

We propose to add an additional loss, called attention consistency loss l_{reg}, to accelerate the training. l_{reg} encourages the multiplication of W_{tts} and W_{asr} close to an identity matrix,

$$l_{reg} = \sum (I - W_{tts} W_{asr})^2 \tag{5}$$

where I denotes the identity matrix. The meaning of this loss is that if the ASR model attends on a specific spectrogram frame to recognize a character, the TTS model is encouraged to attend on the character to generate the specific spectrogram frame.

With l_{reg}, the new ASR loss will become:

$$l'_{asr} = l_{asr} + l_{reg}. \tag{6}$$

This regularization is shown to accelerate the convergence in the experiment.

4.3. Training tricks – Randomness Procedure

When minimizing the ASR loss, there are two choices: sampling a character sequence x^j or sampling a new mel spectrogram y^j. In Algorithm 1, we choose the former one. Remind that we want to prevent the reference encoder to encode information other than prosody or speaker information. Therefore, in each training iteration, we keep the reference mel spectrogram unchanged, and pair the spectrogram with different character sequences. Then ASR loss will keep the reference encoder to encode only prosody information, otherwise, the generated utterance will not be correctly recognized by ASR model. In the experiments, randomly sampling a new character sequence x^j obtains much better performance than sampling a new reference mel spectrogram y^j[1].

5. EXPERIMENTAL SETUP

We used VCTK dataset [35] for speech synthesis, which consisted of around 44000 acoustic utterances with transcriptions and 109 different speaker labels. To save the training time, only the utterances shorter than 3.1 seconds were used, which left around 40000 utterances. We split the training/testing data as 107/2 speakers. While in the training process, the speaker labels were not used. In the training of ASR model, we used the training data of VCTK and part of the LibriSpeech corpus [36], which was a multi-speaker dataset. Also, only the utterances shorter than 3.1 seconds in LibriSpeech were used, resulting in totally around 55000 utterances.

For the TTS and reference embedding, we used the same architecture as in GST-tacotron [4, 8] if not specified. The reduction factor was set to 5. We used the lowercase alphabets as input, and the predicted targets were log-scale mel spectrogram and log-scale linear spectrogram. We first pretrained the ASR for 100 epochs and TTS model for 300 epochs, and then trained the TTS model with the guidance of ASR for 100 epochs. The batch size was set to 32. The learning rate was set to 1e-3 when pretraining ASR and TTS. When training the whole framework, the learning rate was set to 1e-3 and 1e-4 on the TTS loss and ASR loss. Learning rate decay of 0.9 per 4000 steps was used in every case. For fair comparison, we further trained the pretrained TTS model for 100 more epochs as the baseline TTS model without ASR. Therefore, the TTS models with and without ASR have the same training epochs.

6. EXPERIMENTAL RESULT

6.1. Objective Evaluation

For testing, we synthesized 200 sentences with 10 reference utterances, so there were 2000 synthesized utterances in total. The sentences were from LibriSpeech which were not seen in training stage. The reference utterances comprised 5 male and 5 female utterances. The reference utterances were all from different speakers, while 8 from training speakers of VCTK and 2 from testing speakers. Although some of the reference utterances had been seen by the models during training, the sentences to be synthesized had never been seen before the testing.

6.1.1. Attention Consistency Loss

We first compare the training convergence speed of the proposed approach with and without attention consistency loss. Figure 3 shows the accuracy of the end-to-end ASR model used to train the TTS model. The accuracy is evaluated on the speech synthesized from training sentences and reference utterances. During the training, the accuracy became higher and higher, which means that the TTS models learned to synthesize more clear speech that can be correctly recognized by the end-to-end ASR model. With attention consistency loss, the ASR accuracy of the synthesized speech not only converges faster but also becomes higher compares to the one without it. The results demonstrate the effectiveness of attention consistency loss. In the following experiments, we will adopt attention consistency loss if not specified.

6.1.2. ASR Model Attention

Besides ASR accuracy, we also analyze the attention heat map of the end-to-end ASR model during different training stages

[1] We do not show the results in this paper due to space limitation.

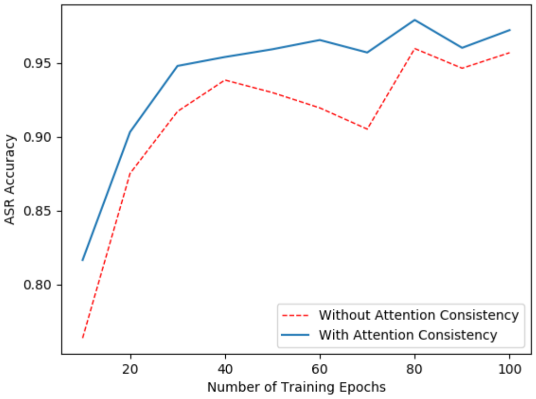
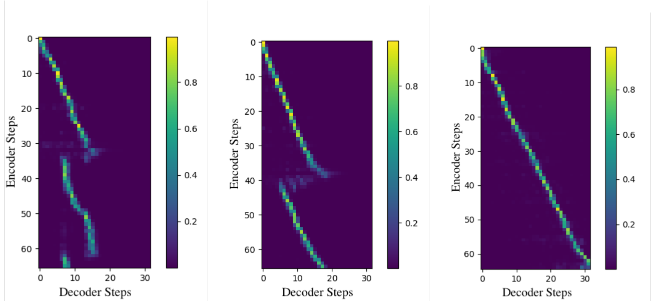


Fig. 3: The accuracy of the end-to-end ASR model on the synthesized speech to the number of training epochs.

of the TTS model. A representative example is shown in Figure 4. It is clear that with the training step increasing, the attention is more and more closer to a diagonal line. Because speech and its transcription have a monotonic mapping relationship [37, 38], the diagonal line in the attention map often signs that the ASR model can successfully recognize the speech.



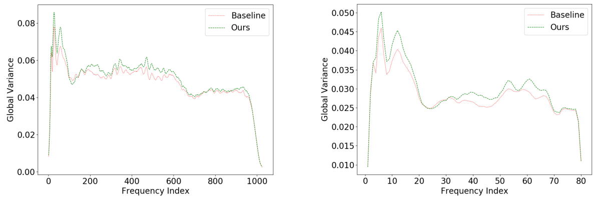
(a) Epoch 0 (b) Epoch 40 (c) Epoch 80

Fig. 4: An example of the attention heat map of the ASR model with different numbers of epochs.

6.1.3. Global variance analysis

Diversified distribution over all frequencies is a highly desired property of synthesized speech signals. This property can be verified by calculating the Global Variance (GV) [39] over the spectrogram. Higher global variance indicates sharpness of the synthesized speech. Figure 5 (a) and (b) show the global variance of the proposed approach (with ASR) and the baseline (without ASR) on linear spectrogram and mel spectrogram respectively.

The results demonstrate that our model has higher global variance almost on all frequency indices compared to the baseline on both linear spectrogram and mel spectrogram.



(a) linear spectrogram (b) mel spectrogram

Fig. 5: The global variance of the generated speech.

Methods		CER	
		Google	**Sphinx**
w/ style token layer	(a)baseline	0.59	0.65
	(b)ours	**0.36**	**0.39**
w/o style token layer	(c)baseline	0.74	0.82
	(d)ours	**0.61**	**0.69**

Table 1: Comparison of the character error rate (CER) of the speech synthesized by different TTS models.

6.1.4. ASR error analysis

To verify that our model can effectively solve the problem of the blurriness of the generated speech, we use off-the-shelf ASR systems, Google and Sphinx [40] systems[2], to recognize the generated speech of different TTS models. The results are shown in Table 1.

In rows (a) and (b), the network architecture of the TTS models is the same as the original GST-tacotron model [9]. We can observe that the character error rate (CER) of our model are lower than that of the baseline of both ASR systems (rows (b) v.s. (a)). In rows (c) and (d), we take away the style token layer from the GST-tacotron, which greatly increases the capacity of the reference encoder. We compare the proposed approach and the baseline based on different TTS models because we want to verify that the proposed approach is independent to the network architecture of the TTS models. The increasing capacity of the reference encoder will let the model more possible to encode text information inside the reference embedding, which will cause the generated speech even more blurry, and thus lead to higher CER (rows (c) v.s. (a)). In this case, adding an additional ASR loss can still mitigate the blurriness problem, and achieve higher accuracy (rows (d) v.s. (c)).

Figure 6 illustrates several spectrogram examples and the ASR recognition results of the generated speech by baseline and our framework. We can see that, the baseline TTS model

[2] These ASR systems are only used to evaluate the performance of the TTS systems. They do not involve in the training process.

is more likely to generate blurry spectrogram, and thus the Google ASR system loses some words or generates repeated words, while our model can mitigate this problem.

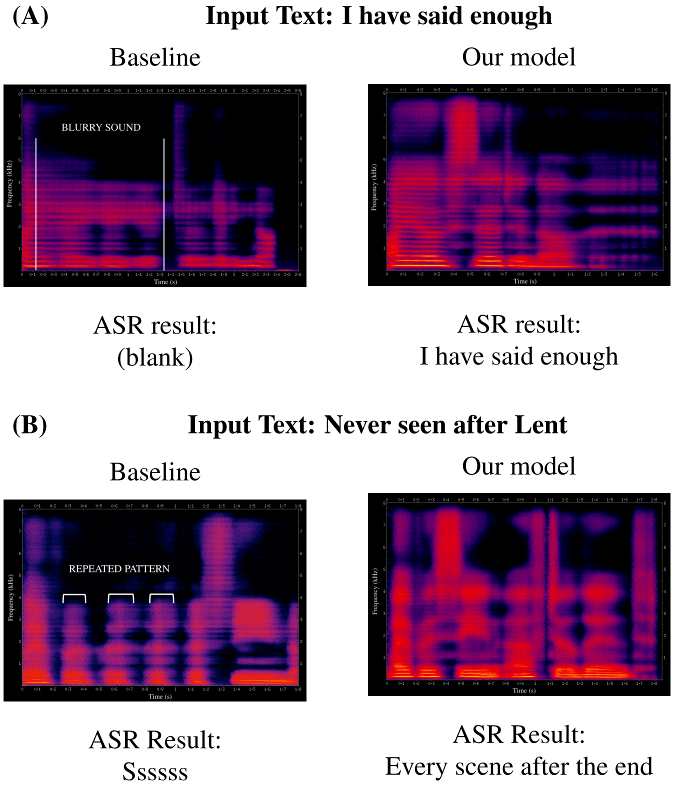


Fig. 6: The spectrograms and the results of the Google ASR system of the generated voice. (A) The generated speech by the baseline model is too blurry to be recognized by the google ASR system. (B) The generated speech by the baseline model has some repeated patterns, which is recognized as many repeated s.

6.1.5. *Speaker Verification Analysis*

To verify whether the TTS models can synthesize the speech of the target speakers, we trained another speaker verification network that took the generated utterances as input to predict the speaker identity. The speaker verification network was trained from all the utterances of ten speakers of the reference utterances in VCTK. The architecture of the speaker verification network was the same architecture as the reference encoder except that the style token layer is replaced with a linear layer. The output dimension of the linear layer was the number of the speakers. The softmax activation was applied on the output of the linear layer. The verification accuracy was 0.747 and 0.739 of our proposed model and the baseline respectively. This shows that while making the synthesized speech more clear, the proposed approach does not make the synthesized speech less similar to the reference utterances.

6.2. Subjective evaluation

We also performed a subjective human evaluation for the generated voices[3]. Four reference utterances, two male and two female utterances, and six sentences were combined mutually to generate twenty-four utterances by both our model and the baseline model. Given one synthesized utterance from our model and one from the baseline model, 15 annotators were asked which one they preferred in terms of two measures: the similarity in speaker prosody to the reference utterance and the clarity of the speech. The utterance pairs were given in random order.

The result is shown in Figure 7. Because the origin GST-tacotron can already model the prosody of the reference utterance well, the result of the similarity is comparable in almost all utterances. That is, for the ability of modeling the speaker characteristics in the reference utterances, the proposed model and the baseline model are basically the same. However, our model performs far better than the baseline in terms of the clarity of the speech. Among the speech generated by our model, around 60% is preferred, and around 22% is comparable. In most of the comparable cases, GST-tacotron had been able to generate clear speech, so the improvement of our model is not significant.

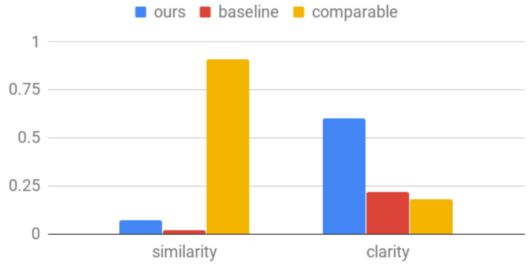


Fig. 7: Results of subjective preference test in terms of the similarity to target speaker (left) and the clarity of the speech (right).

7. CONCLUSION AND FUTURE WORK

In this paper, we proposed a new framework to improve unsupervised style transfer TTS model, GST-tacotron. With the guidance of an end-to-end ASR model, the clarity of the synthesized speech is improved in terms of objective and subjective evaluations. Besides, attention consistency loss is proposed to accelerate the convergence. In our future work, we will use neural-network-based vocoder [41] to further enhance the voice quality. Moreover, we will consider the end-to-end ASR as the discriminator in GAN and update the ASR model while training end-to-end TTS.

[3] Sound demos can be found at `https://happyball.github.io/SLT_demo_page`.

8. REFERENCES

[1] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu, "Wavenet: A generative model for raw audio," *arXiv preprint arXiv:1609.03499*, 2016.

[2] Aaron van den Oord, Yazhe Li, Igor Babuschkin, Karen Simonyan, Oriol Vinyals, Koray Kavukcuoglu, George van den Driessche, Edward Lockhart, Luis C Cobo, Florian Stimberg, et al., "Parallel wavenet: Fast high-fidelity speech synthesis," *arXiv preprint arXiv:1711.10433*, 2017.

[3] Jose Sotelo, Soroush Mehri, Kundan Kumar, Joao Felipe Santos, Kyle Kastner, Aaron Courville, and Yoshua Bengio, "Char2wav: End-to-end speech synthesis," 2017.

[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] Sercan Arik, Gregory Diamos, Andrew Gibiansky, John Miller, Kainan Peng, Wei Ping, Jonathan Raiman, and Yanqi Zhou, "Deep voice 2: Multi-speaker neural text-to-speech," *arXiv preprint arXiv:1705.08947*, 2017.

[6] Andros Tjandra, Sakriani Sakti, and Satoshi Nakamura, "Listening while speaking: Speech chain by deep learning," in *Automatic Speech Recognition and Understanding Workshop (ASRU), 2017 IEEE*. IEEE, 2017, pp. 301–308.

[7] Yaniv Taigman, Lior Wolf, Adam Polyak, and Eliya Nachmani, "Voiceloop: Voice fitting and synthesis via a phonological loop," 2018.

[8] RJ Skerry-Ryan, Eric Battenberg, Ying Xiao, Yuxuan Wang, Daisy Stanton, Joel Shor, Ron J Weiss, Rob Clark, and Rif A Saurous, "Towards end-to-end prosody transfer for expressive speech synthesis with tacotron," *arXiv preprint arXiv:1803.09047*, 2018.

[9] Yuxuan Wang, Daisy Stanton, Yu Zhang, RJ Skerry-Ryan, Eric Battenberg, Joel Shor, Ying Xiao, Fei Ren, Ye Jia, and Rif A Saurous, "Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis," *arXiv preprint arXiv:1803.09017*, 2018.

[10] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," *arXiv preprint*, 2017.

[11] Taeksoo Kim, Moonsu Cha, Hyunsoo Kim, Jung Kwon Lee, and Jiwon Kim, "Learning to discover cross-domain relations with generative adversarial networks," *arXiv preprint arXiv:1703.05192*, 2017.

[12] Fabio Tesser, Giacomo Sommavilla, Giulio Paci, and Piero Cosi, "Experiments with signal-driven symbolic prosody for statistical parametric speech synthesis," in *Eighth ISCA Workshop on Speech Synthesis*, 2013.

[13] Hieu-Thi Luong, Shinji Takaki, Gustav Eje Henter, and Junichi Yamagishi, "Adapting and controlling dnn-based speech synthesis using input codes," in *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*. IEEE, 2017, pp. 4905–4909.

[14] Takashi Nose, Junichi Yamagishi, Takashi Masuko, and Takao Kobayashi, "A style control technique for hmm-based expressive speech synthesis," *IEICE TRANSACTIONS on Information and Systems*, vol. 90, no. 9, pp. 1406–1413, 2007.

[15] Ehsan Variani, Xin Lei, Erik McDermott, Ignacio Lopez Moreno, and Javier Gonzalez-Dominguez, "Deep neural networks for small footprint text-dependent speaker verification," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 4052–4056.

[16] Ye Jia, Yu Zhang, Ron J Weiss, Quan Wang, Jonathan Shen, Fei Ren, Zhifeng Chen, Patrick Nguyen, Ruoming Pang, Ignacio Lopez Moreno, et al., "Transfer learning from speaker verification to multispeaker text-to-speech synthesis," *arXiv preprint arXiv:1806.04558*, 2018.

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

[18] Igor Jauk, "Unsupervised learning for expressive speech synthesis," 2017.

[19] Najim Dehak, Patrick J Kenny, Réda Dehak, Pierre Dumouchel, and Pierre Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, 2011.

[20] George Saon, Hagen Soltau, David Nahamoo, and Michael Picheny, "Speaker adaptation of neural network acoustic models using i-vectors.," in *ASRU*, 2013, pp. 55–59.

[21] Lian-Wu Chen, Wu Guo, and Li-Rong Dai, "Speaker verification against synthetic speech," in *Chinese Spoken Language Processing (ISCSLP), 2010 7th International Symposium on*. IEEE, 2010, pp. 309–312.

[22] Yuki Saito, Shinnosuke Takamichi, and Hiroshi Saruwatari, "Training algorithm to deceive anti-spoofing verification for dnn-based speech synthesis," in *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*. IEEE, 2017, pp. 4900–4904.

[23] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[24] R Kubichek, "Mel-cepstral distance measure for objective speech quality assessment," in *Communications, Computers and Signal Processing, 1993., IEEE Pacific Rim Conference on*. IEEE, 1993, vol. 1, pp. 125–128.

[25] Bajibabu Bollepalli, Lauri Juvela, Paavo Alku, et al., "Generative adversarial network-based glottal waveform model for statistical parametric speech synthesis," 2017.

[26] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.

[27] Daniel Griffin and Jae Lim, "Signal estimation from modified short-time fourier transform," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 32, no. 2, pp. 236–243, 1984.

[28] Chung-Cheng Chiu, Tara N Sainath, Yonghui Wu, Rohit Prabhavalkar, Patrick Nguyen, Zhifeng Chen, Anjuli Kannan, Ron J Weiss, Kanishka Rao, Katya Gonina, et al., "State-of-the-art speech recognition with sequence-to-sequence models," *arXiv preprint arXiv:1712.01769*, 2017.

[29] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, 2017, pp. 5998–6008.

[30] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin, "Convolutional sequence to sequence learning," *arXiv preprint arXiv:1705.03122*, 2017.

[31] William Chan, Navdeep Jaitly, Quoc Le, and Oriol Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 4960–4964.

[32] Navdeep Jaitly, Quoc V Le, Oriol Vinyals, Ilya Sutskever, David Sussillo, and Samy Bengio, "An online sequence-to-sequence model using partial conditioning," in *Advances in Neural Information Processing Systems*, 2016, pp. 5067–5075.

[33] Hasim Sak, Matt Shannon, Kanishka Rao, and Françoise Beaufays, "Recurrent neural aligner: An encoder-decoder neural network model for sequence to sequence mapping," in *Proc. Interspeech*, 2017, pp. 1298–1302.

[34] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton, "Speech recognition with deep recurrent neural networks," in *Acoustics, speech and signal processing (icassp), 2013 ieee international conference on*. IEEE, 2013, pp. 6645–6649.

[35] Christophe Veaux, Junichi Yamagishi, Kirsten MacDonald, et al., "Superseded-cstr vctk corpus: English multi-speaker corpus for cstr voice cloning toolkit," 2016.

[36] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 5206–5210.

[37] Colin Raffel, Minh-Thang Luong, Peter J Liu, Ron J Weiss, and Douglas Eck, "Online and linear-time attention by enforcing monotonic alignments," *arXiv preprint arXiv:1704.00784*, 2017.

[38] Roee Aharoni and Yoav Goldberg, "Sequence to sequence transduction with hard monotonic attention," 2016.

[39] Tomoki Toda and Keiichi Tokuda, "A speech parameter generation algorithm considering global variance for hmm-based speech synthesis," *IEICE TRANSACTIONS on Information and Systems*, vol. 90, no. 5, pp. 816–824, 2007.

[40] Paul Lamere, Philip Kwok, Evandro Gouvea, Bhiksha Raj, Rita Singh, and William Walker, "The cmu sphinx-4 speech recognition system," .

[41] Jonathan Shen, Ruoming Pang, Ron J Weiss, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng Chen, Yu Zhang, Yuxuan Wang, RJ Skerry-Ryan, et al., "Natural tts synthesis by conditioning wavenet on mel spectrogram predictions," *arXiv preprint arXiv:1712.05884*, 2017.