HHS Public Access

Author manuscript

SLT Workshop Spok Lang Technol. Author manuscript; available in PMC 2023 August 11.

Published in final edited form as:

SLT Workshop Spok Lang Technol. 2023 January ; 2022: 920–927. doi:10.1109/slt54892.2023.10022498.

STYLETTS-VC: ONE-SHOT VOICE CONVERSION BY KNOWLEDGE TRANSFER FROM STYLE-BASED TTS MODELS

Yinghao Aaron Li,

Cong Han,

Nima Mesgarani

Department of Electrical Engineering, Columbia University, USA

Abstract

One-shot voice conversion (VC) aims to convert speech from any source speaker to an arbitrary target speaker with only a few seconds of reference speech from the target speaker. This relies heavily on disentangling the speaker's identity and speech content, a task that still remains challenging. Here, we propose a novel approach to learning disentangled speech representation by transfer learning from style-based text-to-speech (TTS) models. With cycle consistent and adversarial training, the style-based TTS models can perform transcription-guided one-shot VC with high fidelity and similarity. By learning an additional mel-spectrogram encoder through a teacher-student knowledge transfer and novel data augmentation scheme, our approach results in disentangled speech representation without needing the input text. The subjective evaluation shows that our approach can significantly outperform the previous state-of-the-art one-shot voice conversion models in both naturalness and similarity.

Index Terms —

Voice conversion; disentangled representations; text-to-speech; transfer learning

1. INTRODUCTION

Voice conversion (VC) is a technique that converts one speaker's voice into another's voice while preserving linguistic and prosodic information such as phonemes and prosody. Recent advances in deep learning have enriched research on one particular type of voice conversion: one-shot voice conversion. This type of voice conversion, also known as any-to-any voice conversion, aims to convert speech from any source speaker to an arbitrary target speaker using only a few seconds of reference audio from the target speaker. To convert an unseen speaker's voice into another speaker's voice unseen during training, the model needs to learn a shared representation of speech across all potential sources and target speakers [1]. Therefore, learning disentangled representations of speech and speaker identity is crucial for successful one-shot voice conversion.

Several techniques have been proposed for learning disentangled representations, including instance normalization [2, 3, 4], vector quantization [3, 5, 6, 7], transfer learning from ASR or TTS models [8, 9, 10, 11, 12], and adversarial training [13, 14]. These methods, albeit effective, do not guarantee that the empirically trained representations contain no source speaker information. VC systems such as Mellotron [15] and Cotatron [16], on the other hand, use phoneme alignment and pitch curve from the source speech and re-synthesize the speech of the target speaker. Since phoneme alignment and normalized pitch curve are largely speaker-agnostic, the re-synthesized speech should only reflect the speech content and prosody of the source audio without leaking any other source-specific information. These TTS-based methods that theoretically guarantee a disentangled representation still suffer from two essential problems. The major drawback of TTS-based models is that this method requires input text or a sequence of phonemes to generate the alignment which limits its potential for applications in real-time inference. Zhang et. al. [10] has made an attempt to address this problem by training an additional mel-spectrogram encoder that produces the same latent representation as the one generated from phoneme alignment and text representation. This is equivalent to training an automatic speech recognition (ASR) system, but as we show here, this way of encoder training is not optimal. Another obstacle endured by the TTS method is the generalization problem. Since the original TTS models are trained to only reconstruct speech from the pitch and phoneme alignment of the source speaker, there is no guarantee that the synthesized speech will sound natural and similar to the target speakers when the input pitch and phoneme alignment are from a different speaker.

In this paper, we present StyleTTS-VC, a non-parallel one-shot voice conversion framework based on StyleTTS [17], a style-based text-to-speech model. We address the aforementioned generalization problems by first training a StyleTTS speech decoder with a cycle consistency loss function and adversarial objectives. We then train a mel-spectrogram encoder to produce representations that reconstruct the decoder output generated using representations from phoneme alignment for all speakers in the training set with both synthesized and real speech as input. Unlike the previous method [10], our proposed technique does not force the encoded representations to be close to the phoneme alignment representations. The subjective human evaluation shows that our model outperforms the previous state-of-the-art one-shot voice conversion model, YourTTS [11], and two other baseline models, AGAIN-VC [4] and VQMIVC [6], for unseen source and target speakers. Moreover, since our model consists of only convolutional layers without non-causal RNN or transformers, our model has the capability to perform real-time inference with a faster-than-real-time vocoder.

Our work makes multiple contributions: (i) we show that the cycle consistency and adversarial objective are effective in training both TTS decoder and mel-spectrogram encoder for VC applications, (ii) we introduce novel data augmentation using text-guided voice conversion results as both input and target during training, and (iii) we demonstrate that the loss function proposed in [10] is suboptimal for transfer learning from TTS models for voice conversion applications and propose an alternative solution with a mutual information (MI) maximization objective. The audio samples from our model are available at https://styletts-vc.github.io.

2. METHODS

2.1. StyleTTS

StyleTTS [17] is a style-based non-autoregressive text-to-speech model that integrates style information through adaptive instance normalization (AdaIN) [18]. The StyleTTS framework consists of eight modules: text encoder, style encoder, discriminator, text aligner, pitch extractor, speech decoder, duration predictor, and prosody predictor. Since we only use the speech decoder for voice conversion, we only describe the modules and objectives needed to train the speech decoder here. We do not use the duration and prosody predictors because they are not relevant to the VC application. For simplicity, we assume that the text aligner and pitch extractor are pre-trained and fixed during training. An overview of StyleTTS decoder training is given in Figure 1a.

Text encoder.—Given input phonemes $\boldsymbol{t}$, our text encoder T encodes $\boldsymbol{t}$ into latent representation $\boldsymbol{h}_{\text{text}} = T(\boldsymbol{t})$. We use the same text encoder as in Tacotron 2 [19].

Style encoder.—Given an input mel-spectrogram $\boldsymbol{x}$, the encoder extracts the style code $s = S(\boldsymbol{x})$. For the voice conversion application, s is roughly equivalent to the speaker embedding. The style encoder is the same as in StarGANv2-VC [20] without the domain-specific linear projection layers.

Decoder.—The decoder G synthesizes the mel-spectrogram $\hat{\boldsymbol{x}} = G(\boldsymbol{h}_{\text{text}} \cdot \boldsymbol{d}_{\text{align}}, s, p_x, n_x)$ from an input audio $\boldsymbol{x}$, where $\boldsymbol{h}_{\text{text}} \cdot \boldsymbol{d}_{\text{align}}$ is the aligned latent representation of phonemes, s is the style code of target speaker, p_x is pitch contour and n_x is the log norm (energy) of $\boldsymbol{x}$ per frame. Our decoder consists of seven residual blocks with AdaIN (equation 1), with which the style code s is introduced into G. The p_x and n_x are normalized and concatenated with the output from every residual block as the input to the next residual block.

$$\text{AdaIN}(x, s) = L_\sigma(s)\frac{x - \mu(x)}{\sigma(x)} + L_\mu(s), \tag{1}$$

where x is a single channel of the feature maps, s is the style vector, $\mu(\cdot)$ and $\sigma(\cdot)$ denotes the channel mean and standard deviation, and L_σ and L_μ are learned linear projections for computing the adaptive gain and bias using the style vector s.

Discriminator.—We employ the same discriminator D as in StarGANv2-VC [20] for seen speakers during training. The discriminator has the same architecture as the style encoder but with the domain-specific linear projection layer for each speaker. The domain-specific layer helps the discriminator to capture detailed features of each speaker in the training set.

Text aligner and pitch extractor.—The text aligner A is based on the decoder of Tacotron 2 with attention. It is pre-trained for automatic speech recognition (ASR) task on the LibriTTS corpus [21]. The pitch extractor F is a pre-trained JDC network [22] trained on LibriTTS with ground truth F0 estimated using Harvest [23]. The text aligner is the same as

the ASR model in [20], and the pitch extractor is the same as the F0 network used in [20]. Both models are pre-trained and fixed during training.

2.2. StyleTTS-VC

For voice conversion without text input, we train an additional encoder E that encodes a mel-spectrogram $\boldsymbol{x}$ into $\boldsymbol{h}_{\text{en}}$ such that $G(\boldsymbol{h}_{\text{text}} \cdot \boldsymbol{d}_{\text{align}}, s, p_{\boldsymbol{x}}, n_{\boldsymbol{x}}) = G(E(\boldsymbol{x}), s, p_{\boldsymbol{x}}, n_{\boldsymbol{x}})$. That is, the encoder learns to produce representations that can be used by the decoder to synthesize the same speech as from the representations generated by the text encoder and phoneme alignment. The encoder consists of six 1-D residual blocks with instance normalization [24], similar to those used in [4] and [20]. Unlike [10], we do not enforce $\boldsymbol{h}_{\text{en}} = \boldsymbol{h}_{\text{text}} \cdot \boldsymbol{d}_{\text{align}}$, in which case the encoder becomes an ASR model and may produce unnatural speech. The effect of enforcing $\boldsymbol{h}_{\text{en}} = \boldsymbol{h}_{\text{text}} \cdot \boldsymbol{d}_{\text{align}}$ is examined in section 3.4.

During inference, for any given input $\boldsymbol{x}_{\text{in}}$, we extract the pitch $p_{\text{in}} = F(\boldsymbol{x}_{\text{in}})$ and energy $n_{\text{in}_t} = \log\sqrt{\sum_{n=1}^{N}(x_{n,t})^2}$ where $\boldsymbol{x}_{n,t}$ represents the n^{th} mel of the t^{th} frame, N the number of mels, and the speech content $\boldsymbol{h}_{\text{en}} = E(\boldsymbol{x}_{\text{in}})$. We compute the style code $s = S(\boldsymbol{x}_{\text{ref}})$ to synthesize $\hat{\boldsymbol{x}}_{\text{trg}}$ from the target speaker. Since both p_{in} and n_{in} are 1-dimensional normalized curves, they cannot contain more information than pitch and volume. Since $\boldsymbol{h}_{\text{en}}$ is trained to replicate the effects of $\boldsymbol{h}_{\text{text}} \cdot \boldsymbol{d}_{\text{align}}$ for all possible speech generated by G, $\boldsymbol{h}_{\text{en}}$ is also a disentangled representation for phonemes that contain no speaker information. An overview of StyleTTS-VC is provided in Figure 1b.

2.3. Training Objectives

We train our model in two steps. We first train the decoder with a cycle consistency loss function, and we then train the encoder with the aforementioned objective with a fixed pre-trained decoder. Given a mel-spectrogram $\boldsymbol{x} \in \mathcal{X}_{y_{\text{src}}}$, a reference $\boldsymbol{x}_{\text{ref}} \in \mathcal{X}_{y_{\text{trg}}}$, the source speaker $y_{\text{src}} \in \mathcal{Y}$ and the target speaker $y_{\text{trg}} \in \mathcal{Y}$, we train our model with the following loss functions.

Mel reconstruction loss.—Given a mel-spectrogram $\boldsymbol{x} \in \mathcal{X}$ and its corresponding text $\boldsymbol{t} \in \mathcal{T}$, the decoder is trained with

$$\mathcal{L}_{rec} = \mathbb{E}_{\boldsymbol{x},t}\left[\left\|\boldsymbol{x} - G(\boldsymbol{h}_{\text{text}} \cdot \boldsymbol{d}_{\text{align}}, s, p_{\boldsymbol{x}}, n_{\boldsymbol{x}})\right\|_1\right], \tag{2}$$

where $\boldsymbol{h}_{\text{text}} = T(\boldsymbol{t})$ is the encoded phoneme representation, $\boldsymbol{d}_{\text{align}}$ is the attention alignment pre-computed from the text aligner, $s = S(\boldsymbol{x})$ is the style code of $\boldsymbol{x}$, $p_{\boldsymbol{x}} = F(\boldsymbol{x})$ is the pitch F0 of $\boldsymbol{x}$ and $n_{\boldsymbol{x}}$ is the energy of $\boldsymbol{x}$. We use the monotonic version of the attention alignment obtained by a dynamic programming algorithm [25] for 50% of the time because the attention alignments are not strictly monotonic and can contain speaker information.

Style reconstruction loss.—To learn meaningful style code that represents speaker embeddings, we used a self-supervised style reconstruction similar to [20]

$$\mathscr{L}_{sty} = \mathbb{E}_{\boldsymbol{x},t,\boldsymbol{x}_{\text{ref}}}\left[\left\|S(\boldsymbol{x}_{\text{ref}}) - S(\hat{\boldsymbol{x}}_{\text{trg}})\right\|_1\right], \tag{3}$$

where $\hat{\boldsymbol{x}}_{\text{trg}} = G(\boldsymbol{h}_{\text{text}} \cdot \boldsymbol{d}_{\text{align}}, S(\boldsymbol{x}_{\text{ref}}), p_{\boldsymbol{x}}, n_{\boldsymbol{x}})$, the reconstructed mel-spectrogram under style code of $\boldsymbol{x}_{\text{ref}}$ with the phonemes, alignment, pitch, and energy information of $\boldsymbol{x}$.

Encoder loss.—When training the encoder, we require the encoder to produce representations that can be used by the decoder to produce the same speech as those generated using representations through phoneme alignment under the encoder loss for an arbitrary target speaker in the training set

$$\mathscr{L}_{en} = \mathbb{E}_{\boldsymbol{x},t,\boldsymbol{x}_{\text{ref}}}\left[\left\|\hat{\boldsymbol{x}} - G(E(\boldsymbol{x}), S(\boldsymbol{x}_{\text{ref}}), p_{\boldsymbol{x}}, n_{\boldsymbol{x}})\right\|_1\right], \tag{4}$$

where $\hat{x} = G(\boldsymbol{h}_{\text{text}} \cdot \boldsymbol{d}_{\text{align}}, S(\boldsymbol{x}_{\text{ref}}), p_{\boldsymbol{x}}, n_{\boldsymbol{x}})$ the converted speech using text representation and phoneme alignment.

Here x can be either ground truth from the training set or synthesized data. $\boldsymbol{x} = G\left(\hat{\boldsymbol{h}}_{\text{text}} \cdot \hat{\boldsymbol{d}}_{\text{align}}, S(\hat{\boldsymbol{x}}_{\text{ref}}), p_{\hat{\boldsymbol{x}}}, n_{\hat{\boldsymbol{x}}}\right)$ when $\boldsymbol{x}$ is synthesized, where $\hat{\boldsymbol{h}}_{\text{text}}$ and $\hat{\boldsymbol{d}}_{\text{align}}$ are text and alignment of another speech sample $\hat{\boldsymbol{x}}$ and $\hat{\boldsymbol{x}}_{\text{ref}}$ is another reference audio different from $\boldsymbol{x}_{\text{ref}}$. That is, when $\boldsymbol{x}$ is synthesized, it is a converted speech sample used as an input. This novel data augmentation fully explores the input and target space of the pre-trained TTS decoder and produces more robust models compared to those trained without this technique.

Phoneme loss.—Since we do not demand $E(\boldsymbol{x}) = \boldsymbol{h}_{\text{text}} \cdot \boldsymbol{d}_{\text{align}}$, there is no guarantee that the generated speech keeps the original phoneme content. We employ a phoneme loss function to maximize the mutual information (MI) [26] between the encoded representations and the phonetic content through a linear projection $\boldsymbol{P}$ for each frame of the input

$$\mathscr{L}_{MI} = \mathbb{E}_{\boldsymbol{x},t}\left[\frac{1}{T}\sum_{i=1}^{T}\mathbf{CE}\left((\boldsymbol{d}_{\text{align}} \cdot \boldsymbol{t})_i, (\boldsymbol{P} \cdot E(\boldsymbol{x}))_i\right)\right], \tag{5}$$

where T is the number of frames and $\mathbf{CE}(\cdot)$ denotes the cross-entropy loss function.

Cycle consistency loss.—To make sure that the decoder generalizes to different input style codes independent of text, pitch and energy, we also employ a cycle consistency loss function

$$\mathscr{L}_{cycle} = \mathbb{E}_{\boldsymbol{x},t}\left[\left\|\boldsymbol{x} - G\left(\boldsymbol{h}, S(\boldsymbol{x}), p_{\hat{\boldsymbol{x}}_{\text{trg}}}, n_{\hat{\boldsymbol{x}}_{\text{trg}}}\right)\right\|_1\right], \tag{6}$$

where $p_{\hat{\boldsymbol{x}}_{\text{trg}}}$ is the pitch curve and $n_{\hat{\boldsymbol{x}}_{\text{trg}}}$ is the energy of the converted speech $\hat{\boldsymbol{x}}_{\text{trg}}$. When training the decoder, $\boldsymbol{h} = \boldsymbol{h}_{\text{text}} \cdot \hat{\boldsymbol{d}}_{\text{align}}$. and $\boldsymbol{x}$ is the ground truth where $\hat{\boldsymbol{d}}_{\text{align}}$ is the attention alignment of the converted speech $\hat{x}_{\text{trg}}$. When training the encoder, $\boldsymbol{h} = E(\boldsymbol{x})$ and $\boldsymbol{x} = \hat{\boldsymbol{x}}$ in equation 4.

Adversarial loss.—We use two adversarial objectives: the original cross-entropy loss function for adversarial training following [20] and the additional feature-matching loss following [27]

$$\mathcal{L}_{adv} = \mathbb{E}_{\boldsymbol{x}, y_{\text{src}}}[\log D(\boldsymbol{x}, y_{\text{src}})] + \mathbb{E}_{\boldsymbol{x}, t, y_{\text{trg}}}[\log(1 - D(\hat{\boldsymbol{x}}_{\text{trg}}, y_{\text{trg}}))], \tag{7}$$

$$\mathcal{L}_{fm} = \mathbb{E}_{\boldsymbol{x}, \hat{\boldsymbol{x}}}\left[\sum_{l=1}^{L} \frac{1}{N_l} \left\| D^l(\boldsymbol{x}, y_{\text{src}}) - D^l(\hat{\boldsymbol{x}}, y_{\text{src}}) \right\|_1\right], \tag{8}$$

where $D(\cdot, y)$ denotes the output of discriminator for the speaker $y \in \mathcal{Y}$, $\hat{\boldsymbol{x}}_{\text{trg}} = G(\boldsymbol{h}, S(\boldsymbol{x}_{\text{ref}}), p_{\boldsymbol{x}}, n_{\boldsymbol{x}})$ the converted speech, $\hat{\boldsymbol{x}} = G(\boldsymbol{h}, s, p_{\boldsymbol{x}}, n_{\boldsymbol{x}})$ the reconstructed speech, L is the total number of layers in D and D^l denotes the output feature map of l^{th} layer with N_l features. The values of $\boldsymbol{x}$ and $\boldsymbol{h}$ are the same as in equation 6 depending on whether the encoder E or the decoder D is trained.

Full objectives.—Our full objective functions for training the decoder can be summarized as follows:

$$\min_{G,T,S} \max_{D} \mathcal{L}_{rec} + \lambda_{sty}\mathcal{L}_{sty} + \lambda_{cycle}\mathcal{L}_{cycle} + \lambda_{adv}\mathcal{L}_{adv} + \lambda_{fm}\mathcal{L}_{fm}, \tag{9}$$

and full objective functions for the encoder are:

$$\min_{E,P} \max_{D} \mathcal{L}_{en} + \lambda_{cycle}\mathcal{L}_{cycle} + \lambda_{MI}\mathcal{L}_{MI} + \lambda_{adv}\mathcal{L}_{adv} + \lambda_{fm}\mathcal{L}_{fm}\,. \tag{10}$$

3. EXPERIMENTS

3.1. Datasets

We used the VCTK [28] corpus to evaluate our models. The VCTK dataset consists of 109 native English speakers with various accents, each of which reads approximately 400 sentences. We followed the same procedure described in [6], where the 89 speakers were randomly selected for training and the rest 20 speakers were used as unseen speakers for testing. We further divided samples of the selected 89 speakers into training and validation sets with a 90%/10% split. The samples were downsampled to 24 kHz. We converted the text sequences into phoneme sequences using an open-source tool[1]. We extracted mel-spectrograms with a FFT size of 2048, hop size of 300, and window length of 1200 in 80 mel bins using TorchAudio [29]. The generated mel-spectrogram was converted into waveforms using the Hifi-GAN [27] and downsampled to 16 kHz to match the baseline models.

3.2. Training Details

We first trained the decoder for 100 epochs with $\lambda_{sty} = 0.2$, $\lambda_{cycle} = 1$, $\lambda_{adv} = 1$ and $\lambda_{fm} = 0.2$, and we then trained the encoder for 100 epochs with $\lambda_{MI} = 1$. We trained both models using the AdamW optimizer [30] with $\beta_1 = 0, \beta_2 = 0.99$, weight decay $\lambda = 10^{-4}$, learning rate $\gamma = 10^{-4}$

[1] https://github.com/Kyubyong/g2p

and batch size of 64 samples. We randomly divided mel-spectrograms into segments of the shortest length in the batch.

3.3. Evaluations

We conducted subjective evaluations with two metrics: the mean opinion score of naturalness (MOS-N) which measures the naturalness of converted speech and the mean opinion score of similarity (MOS-P) which evaluates the similarity between converted and reference speech. We recruited native English speakers located in the U.S. to participate in our evaluations using an online survey through Amazon Mechanical Turk. We compared our model with two recent baseline models, AGAIN-VC [4] and VQMIVC [6], and one state-of-the-art model, YourTTS[11], for any-to-any voice conversion. All baseline models were trained with official implementation[2][3][4] using the same train and test speaker split. For a fair comparison, the mel-spectrograms converted from all models were synthesized with HifiGAN [27] and downsampled to 16 kHz in our evaluations.

In every experiment, we randomly selected 40 sets of samples. When evaluating each set, we randomly permuted the order of the models and instructed the subjects to rate them without revealing the model labels. For each set, we required that there were at least five different speakers reading the same sentence, in which one was used as the ground truth and the rest four were used as the source input for our model and the three baseline models. This ensures that different samples have different lengths so that raters do not find out which one is the ground truth. The method is similar to multiple stimuli with hidden reference and anchor (MUSHRA), enabling the subjects to compare the subtle difference among models. We used the subjective rating of the ground truth as an attention check: all ratings from a subject were dropped from our analyses if the MOS of the ground truth was not ranked the highest among all models. Each set was rated by 10 raters after disqualified raters were dropped.

In addition to subjective evaluations, we also performed objective evaluations using speaker classification and phoneme error rate (PER) from an ASR model to evaluate the speaker similarity and speech intelligibility [20]. The speaker classification model consists of a ResNet-18 network that takes a mel-spectrogram to predict the speaker label. The model was trained on the test speakers and we report the classification accuracy (ACC) of the trained models on samples generated with different models. We converted speech waveforms to text using an ASR model from ESPNet [31] and converted the text to phoneme sequences to calculate PER.

3.4. Ablation Study

To demonstrate that our approach to addressing problems in TTS-based methods is effective, we conducted an ablation study with both subjective and objective evaluations described in section 3.3. We ablated $\mathscr{L}_{MI}$ and $\mathscr{L}_{cycle}$ when training the encoder and the decoder, respectively. In addition, to show that the latent loss introduced in Zhang et. al. [10] hurts the performance, we have added the loss for the encoder training defined as

[2] https://github.com/KimythAnly/AGAIN-VC
[3] https://github.com/Wendison/VQMIVC
[4] https://github.com/Edresson/YourTTS

$$\mathscr{L}_{latent} = \mathbb{E}_{\boldsymbol{x},t}[\|(\boldsymbol{d}_{\text{align}} \cdot \boldsymbol{h}_{\text{text}}) - \boldsymbol{E}(\boldsymbol{x})\|_1], \tag{11}$$

in which the latent representation produced by E is forced to be the same as that generated through text encoder and phoneme alignment. We refer to this case as $+\mathscr{L}_{latent}$ in Table 2. To demonstrate that the data augmentation for the encoder loss is effective, we also trained a model without the data augmentation. That is, we set $\hat{\boldsymbol{x}} = \boldsymbol{x}$ and only use ground truth from the training set as $\boldsymbol{x}$ in equation 4.

4. RESULTS

Table 1 and 2 show the results of comparison between different models and the ground truth. Our model significantly outperforms the other baseline models in both naturalness and similarity in the subjective evaluation experiment. Our model also scored higher in testing accuracy and PER than other baseline models except for YourTTS in PER. However, we do note that the difference is small as shown in Table 2. In addition, since our model is not Flow-based, we do not need to compute the Jacobian and matrix inversion required by Flowbased YourTTS. This makes our model significantly faster than YourTTS as indicated by RTF in Table 2. The ablation study results in Table 3 and 4 show that removing $\mathscr{L}_{MI}$ or $\mathscr{L}_{cycle}$ decreases both naturalness and similarity of the synthesized speech. The baseline model with full objectives also outperforms models trained without $\mathscr{L}_{MI}$ or $\mathscr{L}_{cycle}$ in classification accuracy and PER. Training without data augmentation also decreases the rated naturalness and objective metrics.

It is worth noting that the rated naturalness and similarity drop significantly when we add the proposed $\mathscr{L}_{latent}$ in [10]. We hypothesize that by enforcing $\boldsymbol{d}_{\text{align}} \cdot \boldsymbol{h}_{\text{text}} = \boldsymbol{E}(\boldsymbol{x})$ through $\mathscr{L}_{latent}$, we essentially obtain an ASR model because E is trained to produce an aligned version of $\boldsymbol{h}_{\text{text}}$ which consists of merely phoneme token embeddings. We show an example of inverted alignment using $\boldsymbol{E}(\boldsymbol{x})$ to illustrate our hypothesis. As shown in Figure 2, the inverted alignment $\boldsymbol{E}(\boldsymbol{x}) \cdot \boldsymbol{h}_{\text{text}}^{-1}$ successfully reconstructs the monotonic alignment d_{align} with some noise when E is trained with $\mathscr{L}_{latent}$. This shows that $\boldsymbol{E}(\boldsymbol{x})$ consists roughly of discretized phoneme representations as d_{align} can be recovered through a pseudoinverse of $\boldsymbol{h}_{\text{text}}$. On the other hand, the encoder trained without $\mathscr{L}_{text}$ fails to recover d_{align} with $\boldsymbol{h}_{\text{text}}^{-1}$, indicating that E learns a different representation that the decoder can use to reconstruct the natural speech produced by monotonic alignment and text representation. Training without $\mathscr{L}_{latent}$ avoids the problems associated with ASR models such as incorrectly recognized phonemes that can make speech unclear or produce incorrect phonetic content.

5. CONCLUSIONS

We propose a framework using a style-based TTS model for one-shot voice conversion with novel cycle consistency and phoneme MI maximization objectives in place of the latent reconstruction objective. The framework employs a novel data augmentation scheme that fully explores the input and output space of pre-trained TTS decoders. The proposed model achieves state-of-the-art performance in similarity and naturalness with both subjective and objective evaluations where our models scored significantly higher in various metrics,

including MOS, ACC, and PER than previous models. We also demonstrate that using the latent reconstruction loss $\mathscr{L}_{latent}$ proposed in [10] worsens speech similarity and naturalness and we illustrate a potential explanation for this effect. Moreover, unlike other one-shot voice conversion systems such as [6] and [10], our framework is completely convolutional and can therefore perform real-time inference with a faster-than-real-time vocoder. However, we acknowledge that our framework requires text labels during training, which can be prohibitive to train on unannotated large-scale speech corpora. Future work includes removing the need for text labels through semi-supervised or self-supervised learning. We would also like to improve speaker similarity by learning a better speaker representation that can reproduce the accent of unseen speakers.

ACKNOWLEDGMENTS

This work was funded by the national institute of health (NIH-NIDCD) and a grant from Marie-Josee and Henry R. Kravis.

REFERENCES

[1]. Qian Kaizhi, Zhang Yang, Chang Shiyu, Yang Xuesong, and Hasegawa-Johnson Mark, “Autovc: Zero-shot voice style transfer with only autoencoder loss,” in International Conference on Machine Learning. PMLR, 2019, pp. 5210–5219.

[2]. Chou Ju-chieh, Yeh Cheng-chieh, and Lee Hung-yi, “One-shot voice conversion by separating speaker and content representations with instance normalization,” arXiv preprint arXiv:1904.05742, 2019.

[3]. Wu Da-Yi and Lee Hung-yi, “One-shot voice conversion by vector quantization,” in ICASSP 2020–2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2020, pp. 7734–7738.

[4]. Chen Yen-Hao, Wu Da-Yi, Wu Tsung-Han, and Lee Hung-yi, “Again-vc: A one-shot voice conversion using activation guidance and adaptive instance normalization,” in ICASSP 2021–2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2021, pp. 5954–5958.

[5]. van Niekerk Benjamin, Nortje Leanne, and Kamper Herman, “Vector-quantized neural networks for acoustic unit discovery in the zerospeech 2020 challenge,” arXiv preprint arXiv:2005.09409, 2020.

[6]. Wang Disong, Deng Liqun, Yeung Yu Ting, Chen Xiao, Liu Xunying, and Meng Helen, “Vqmivc: Vector quantization and mutual information-based unsupervised speech representation disentanglement for one-shot voice conversion,” arXiv preprint arXiv:2106.10132, 2021.

[7]. Tang Huaizhen, Zhang Xulong, Wang Jianzong, Cheng Ning, and Xiao Jing, “Avqvc: One-shot voice conversion by vector quantization with applying contrastive learning,” arXiv preprint arXiv:2202.10020, 2022.

[8]. Li Zhonghao, Tang Benlai, Yin Xiang, Wan Yuan, Shen Ling Chen, and Ma Zejun, “Ppg-based singing voice conversion with adversarial representation learning,” in ICASSP 2021–2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2021, pp. 7073–7077.

[9]. Lin Jheng-hao, Lin Yist Y, Chien Chung-Ming, and Lee Hung-yi, “S2vc: A framework for any-to-any voice conversion with self-supervised pretrained representations,” arXiv preprint arXiv:2104.02901, 2021.

[10]. Zhang Mingyang, Zhou Yi, Zhao Li, and Li Haizhou, “Transfer learning from speech synthesis to voice conversion with non-parallel training data,” IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 29, pp. 1290–1302, 2021.

[11]. Casanova Edresson, Weber Julian, Shulby Christopher D, Candido Arnaldo Junior, Gölge Eren, and Ponti Moacir A, “Yourtts: Towards zero-shot multi-speaker tts and zero-shot voice

conversion for everyone," in International Conference on Machine Learning. PMLR, 2022, pp. 2709–2720.

[12]. Gabryś Adam, Huybrechts Goeric, Ribeiro Manuel Sam, Chien Chung-Ming, Roth Julian, Comini Giulia, Barra-Chicote Roberto, Perz Bartek, and Lorenzo-Trueba Jaime, "Voice filter: Few-shot text-to-speech speaker adaptation using voice conversion as a post-processing module," in ICASSP 2022–2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2022, pp. 7902–7906.

[13]. Wang Ruobai, Ding Yu, Li Lincheng, and Fan Changjie, "One-shot voice conversion using star-gan," in ICASSP 2020–2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2020, pp. 7729–7733.

[14]. Tang Huaizhen, Zhang Xulong, Wang Jianzong, Cheng Ning, Zeng Zhen, Xiao Edward, and Xiao Jing, "Tgavc: Improving autoencoder voice conversion with text-guided and adversarial training," in 2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU). IEEE, 2021, pp. 938–945.

[15]. Valle Rafael, Li Jason, Prenger Ryan, and Catanzaro Bryan, "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.

[16]. Park Seung-won, Kim Doo-young, and Joe Myun-chul, "Cotatron: Transcription-guided speech encoder for any-to-many voice conversion without parallel data," arXiv preprint arXiv:2005.03295, 2020.

[17]. Li Yinghao Aaron, Han Cong, and Mesgarani Nima, "Styletts: A style-based generative model for natural and diverse text-to-speech synthesis," arXiv preprint arXiv:2205.15439, 2022.

[18]. Huang Xun and Belongie Serge, "Arbitrary style transfer in real-time with adaptive instance normalization," in Proceedings of the IEEE international conference on computer vision, 2017, pp. 1501–1510.

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

[20]. Li Yinghao Aaron, Zare Ali, and Mesgarani Nima, "Starganv2-vc: A diverse, unsupervised, non-parallel framework for natural-sounding voice conversion," in Interspeech, 2021.

[21]. Zen Heiga, Dang Viet, Clark Rob, Zhang Yu, Weiss Ron J, Jia Ye, Chen Zhifeng, and Wu Yonghui, "Libritts: A corpus derived from librispeech for text-to-speech," arXiv preprint arXiv:1904.02882, 2019.

[22]. Kum Sangeun and Nam Juhan, "Joint detection and classification of singing voice melody using convolutional recurrent neural networks," Applied Sciences, vol. 9, no. 7, pp. 1324, 2019.

[23]. Morise Masanori et al. , "Harvest: A high-performance fundamental frequency estimator from speech signals.," in INTERSPEECH, 2017, pp. 2321–2325.

[24]. Ulyanov Dmitry, Vedaldi Andrea, and Lempitsky Victor, "Instance normalization: The missing ingredient for fast stylization," arXiv preprint arXiv:1607.08022, 2016.

[25]. Kim Jaehyeon, Kim Sungwon, Kong Jungil, and Yoon Sungroh, "Glow-tts: A generative flow for text-to-speech via monotonic alignment search," Advances in Neural Information Processing Systems, vol. 33, 2020.

[26]. Boudiaf Malik, Rony Jérôme, Ziko Imtiaz Masud, Granger Eric, Pedersoli Marco, Piantanida Pablo, and Ayed Ismail Ben, "A unifying mutual information view of metric learning: cross-entropy vs. pairwise losses," in European conference on computer vision. Springer, 2020, pp. 548–564.

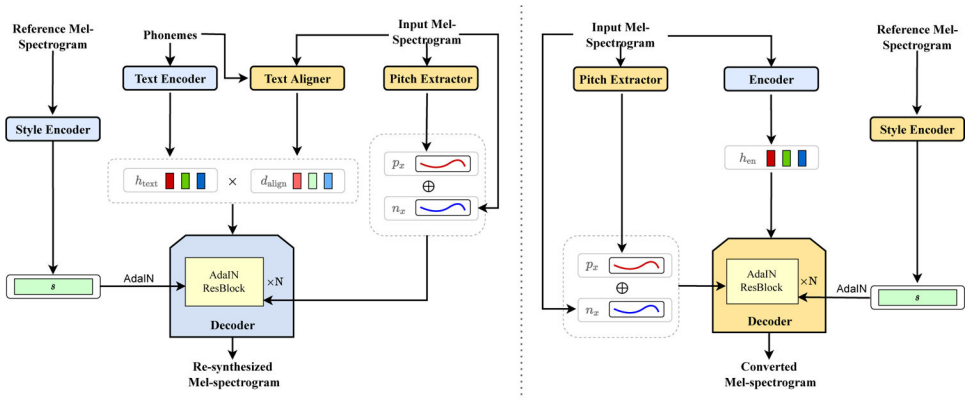
[27]. Kong Jungil, Kim Jaehyeon, and Bae Jaekyoung, "Hifigan: Generative adversarial networks for efficient and high fidelity speech synthesis," Advances in Neural Information Processing Systems, vol. 33, 2020.

[28]. Yamagishi Junichi, Veaux Christophe, MacDonald Kirsten, et al., "Cstr vctk corpus: English multi-speaker corpus for cstr voice cloning toolkit (version 0.92)," 2019.

[29]. Yang Yao-Yuan, Hira Moto, Ni Zhaoheng, Chourdia Anjali, Astafurov Artyom, Chen Caroline, Yeh ChingFeng, Puhrsch Christian, Pollack David, Genzel Dmitriy, Greenberg Donny, Yang Edward Z., Lian Jason, Mahadeokar Jay, Hwang Jeff, Chen Ji, Goldsborough Peter, Roy Prabhat, Narenthiran Sean, Watanabe Shinji, Chintala Soumith, Quenneville-Bélair Vincent, and Shi Yangyang, "Torchaudio: Building blocks for audio and speech processing," arXiv preprint arXiv:2110.15018, 2021.

[30]. Loshchilov Ilya and Hutter Frank, "Fixing weight decay regularization in adam," 2018.

[31]. Watanabe Shinji, Hori Takaaki, Karita Shigeki, Hayashi Tomoki, Nishitoba Jiro, Unno Yuya, Soplin Nelson Enrique Yalta, Heymann Jahn, Wiesner Matthew, Chen Nanxin, Renduchintala Adithya, and Ochiai Tsubasa, "ESPnet: End-to-end speech processing toolkit," in Proceedings of Interspeech, 2018, pp. 2207–2211.



(a) Decoder training (b) Encoder training and inference

Fig. 1.

Training and inference schemes for StyleTTS-VC. The modules in blue are trained while those in orange are pretrained and hence fixed during training. (a) Step 1 of training where the decoder is trained to synthesize target speech from a reference mel-spectrogram and pitch curve, energy, phoneme alignment and text from an input mel-spectrogram. (b) Step 2 of training and inference procedures where the text aligner and text encoder are replaced by a mel-spectrogram encoder.

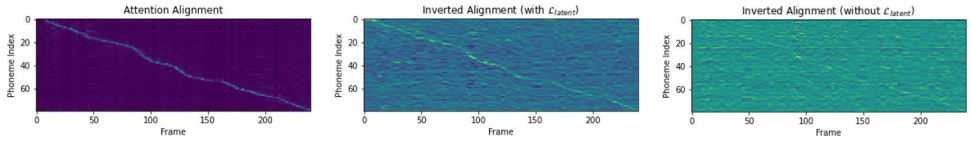


Fig. 2.

Example of attention alignment d_{align} and inverted alignments obtained through $E(\boldsymbol{x}) \cdot \boldsymbol{h}_{\text{text}}^{-1}$ where $\boldsymbol{h}_{\text{text}}^{-1}$ is a pseudoinverse of $\boldsymbol{h}_{\text{text}}$. The representation trained with $\mathscr{L}_{latent}$ clearly reproduces the monotonic alignment, indicating that E acts like an ASR model.

Table 1.

Comparison of MOS with 95% confidence intervals between different models.

Method	MOS-N	MOS-P
Ground Truth	4.68 (± 0.05)	4.58 (± 0.07)
StyleTTS-VC	**3.89** (± **0.09**)	**3.66** (± **0.10**)
YourTTS	3.70 (± 0.10)	3.45 (± 0.10)
VQMIVC	2.85 (± 0.09)	2.50 (± 0.10)
AGAIN-VC	2.11 (± 0.08)	2.16 (± 0.10)

Table 2.

Objective evaluation results of test accuracy (ACC), phoneme error rate (PER) and real time factor (RTF) between different models. The RTF was calculated under a single NVIDIA GeForce RTX 3090 Ti GPU.

Method	ACC ↑	PER ↓	RTF ↓
Ground Truth	100%	2.9%	–
StyleTTS-VC	**91.7%**	6.17%	0.0128
YourTTS	49.4 %	**5.58%**	0.0369
VQMIVC	36.0%	26.0%	**0.0115**
AGAIN-VC	70.0%	24.6%	0.0143

Table 3.

Subjective evaluation results of mean opinion scores (MOS) with 95% confidence intervals (CI) between different training objectives.

Method	MOS-N	MOS-P
Proposed	**3.85** (± **0.09**)	**3.67** (± **0.11**)
w/o augmentation	3.78 (± 0.09)	3.67 (± 0.12)
$-\mathscr{L}_{MI}$	3.74 (± 0.10)	3.63 (± 0.12)
$-\mathscr{L}_{Cycle}$	3.70 (± 0.10)	3.62 (± 0.11)
$+\mathscr{L}_{latent}$	3.60 (± 0.10)	3.58 (± 0.11)

Table 4.

Objective evaluation results of speaker-classification test accuracy (ACC) and phoneme error rate (PER) between different training objectives.

Method	ACC ↑	PER ↓
Baseline	**91.7%**	**10.4%**
w/o augmentation	90.5%	11.5%
$-\mathscr{L}_{MI}$	91.4%	14.9%
$-\mathscr{L}_{Cycle}$	90.0%	17.1%
$+\mathscr{L}_{latent}$	91.2%	20.6%