Title

Robust Disentangled Variational Speech Representation Learning for Zero-Shot Voice Conversion

Authors

Lian, Jiachen
Zhang, Chunlei
Yu, Dong

Publication Date

2022-05-27

DOI

10.1109/icassp43922.2022.9747272

ROBUST DISENTANGLED VARIATIONAL SPEECH REPRESENTATION LEARNING FOR ZERO-SHOT VOICE CONVERSION

Jiachen Lian[1,2]*, Chunlei Zhang*[2]*, Dong Yu*[2]

[1] UC Berkeley, EECS, CA [2] Tencent AI Lab, Bellevue, WA

jiachenlian@berkeley.edu, {cleizhang, dyu}@tencent.com

ABSTRACT

Traditional studies on voice conversion (VC) have made progress with parallel training data and known speakers. Good voice conversion quality is obtained by exploring better alignment modules or expressive mapping functions. In this study, we investigate zero-shot VC from a novel perspective of self-supervised disentangled speech representation learning. Specifically, we achieve the disentanglement by balancing the information flow between global speaker representation and time-varying content representation in a sequential variational autoencoder (VAE). A zero-shot voice conversion is performed by feeding an arbitrary speaker embedding and content embeddings to the VAE decoder. Besides that, an on-the-fly data augmentation training strategy is applied to make the learned representation noise invariant. On TIMIT and VCTK datasets, we achieve state-of-the-art performance on both objective evaluation, i.e., speaker verification (SV) on speaker embedding and content embedding, and subjective evaluation, i.e., voice naturalness and similarity, and remains to be robust even with noisy source/target utterances.

***Index Terms*—** Self-supervised Disentangled representation learning, zero-shot style transfer, voice conversion, variational autoencoder

1. INTRODUCTION

Voice Conversion (VC) seeks to automatically convert the non-linguistic information of a source speaker to a target speaker, while keeping the linguistic content unchanged. The non-linguistic information may include timbre (i.e., speaker identity), emotion, accent or rhythm, to name a few. Due to its potential for applications in privacy protection, security and entertainment industry etc. [1, 2, 3], VC has received long-term research interest.

We can categorise current VC systems into two methodologies. The first one employs a conversion model to map source acoustic features to target acoustic features [1,4,5] . For the conventional VC approaches with parallel training data, acoustic features are first extracted from the source and target utterances. Then, the acoustic features are aligned frame-wise with an alignment module [6]. Studies have shown that the alignment step can be bypassed through using sequence-to-sequence models for direct source-target acoustic mapping with better VC performance [7]. For direct mapping VC with nonparallel training data, progress has been made with generative adversarial networks (GAN) based many-to-many VC systems [8,9]. Although widely investigated, the direct mapping method assumes that the speaker of source-target VC pair is pre-known, which limits the application of such models in the real world. To relax this constraint, the second methodology constructs VC based on explicitly learned speaking style and content representations. Among these approaches, phonetic posteriorgrams (PPGs) is widely used as the speaker independent content representations [10, 11], and speaker embeddings extracted from a pre-trained speaker verification model are often assumed to carry timbre information [12]. They have been successfully applied to tasks such as many-to-many VC or any-to-many VC. For zero-shot VC, both AUTOVC and AdaIN-VC construct encoder-decoder frameworks [13, 14]. The encoder compress the speaking style and the content information into the latent embedding, and the decoder generate a voice sample by combining a speaking style embedding and a content embedding. To achieve a better VC performance, these models require positive pair of utterances (i.e., two utterances come from the same speaker) during training, and the systems have to rely on pre-trained speaker models.

In this study, we focus on the problem of speaker identity conversion. We extend VAE as the backbone framework for learning disentangled content representation and speaking style representation, where balanced content and style information flow is achieved in the VAE training [15]. We show that the vanilla VAE [15, 16] loss can be extended to force strong disentanglement between speaker and content components, which is intuitively explained from three levels. In addition, we explore to make the learned representation robust against background noise/music and interfering speaker etc. An on-the-fly data augmentation is introduced as the inductive bias to the VAE training, with this training strategy, we arrive at a denoising disentangled sequential VAE (D-DSVAE), where low quality speech input is allowed to test for VC. With all these contributions, our proposed system achieves the state-of-the-art VC performance and improved robustness.

2. PROPOSED METHODS

We start with introducing some notations. Denote a speech segment variable $X = [x_1, x_2, ..., x_T]$, which is the STFT or Mel-Spectrogram in our implementation. Denote $Z_S \in \mathbb{R}^{d_S}$ as speaker style latent representation and $Z_C = [z_{c1}, z_{c2}, ..., z_{cT}] \in \mathbb{R}^{d_C \times T}$ as speech content latent representations. Denote θ as model parameters. The proposed VC system adopts the modified form of DSVAE [16] as our backbone, which is shown in Fig.1. The model takes X as input, which is first passed into a shared encoder E_{share} and it gives $W = [w_1, w_2, ..., w_T] \in \mathbb{R}^{d_W \times T}$. Here d_S, d_C and d_W are positive integers. Then a speaker encoder $E_{speaker}$ and a content encoder $E_{content}$ take W as input and model the posterior distribution $q_\theta(Z_S|W) = q_\theta(Z_S|X)$ and $q_\theta(Z_C|W) = q_\theta(Z_C|X)$ respectively. Z_S and Z_C are then obtained via sampling from $q(Z_S|X)$ and $q(Z_C|X)$. The expectation is that Z_S only encodes speaker information and Z_C only encodes content information. During the generation stage, the concatenation of Z_S and Z_C are passed into a shared decoder D_{share} to generate the spectrogram $\hat{X}$, i.e. $\hat{X} = D(Z_S, Z_C)$. A vocoder is then applied to convert $\hat{X}$ to waveform.

Work done when Jiachen was an intern at Tencent AI Lab, Bellevue, WA

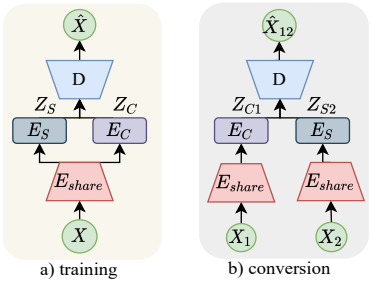

Fig. 1. A flow diagram of proposed VC system.

When performing voice conversion from source X_1 to target X_2, and the converted speech is $\hat{X}_{12} = D(Z_{S2}, Z_{C1})$, where Z_{S2} and Z_{C1} are sampled from $q(Z_S|X_2)$ and $q(Z_C|X_1)$ respectively, as shown in Fig.1. In the following sections, we first highlight the probabilistic graphic models in Sec.2.1. Then, the objective function is introduced in Sec.2.2, then we discuss its validity via proposing three sufficient conditions for achieving disentanglement between Z_S and Z_C.

2.1. Disentanglement-Aware Probabilistic Graphical Models

The frequently used method to achieve disentanglement between two components is to let these two components be probabilistically independent with each other. Following such intuition, we factorize both the prior distributions and posterior distributions of Z_S and Z_C by following independence assumption, which is consistent with [16, 17]. We denote $Z = [Z_S, Z_C]$ as a joint latent representation.

Prior The joint prior distribution is factorised as follows:

$$p_\theta(Z) = p(Z_S)p_\theta(Z_C) = p(Z_S)\prod_{t=1}^{T} P_\theta(z_{ct}|z_{<t}) \tag{1}$$

where $p(Z_S)$ is a standard normal distribution and $p_\theta(Z_C)$ is modeled by an autoregressive LSTM.

Posterior The joint posterior distribution is factorised as follows:

$$q_\theta(Z|X) = q_\theta(Z_S, Z_C|W) = q_\theta(Z_S|W)q_\theta(Z_C|W) \tag{2}$$

Here, $q_\theta(Z_S|W)$ and $q_\theta(Z_C|W)$ are modeled by two independent LSTMs. Implementation Detailes can be found in 3.3.

2.2. Loss Objectives

We first provide the loss objective as shown in Eq. 3, which is simply the vanilla VAE [15] loss. After that, we will explain how it is related to disentanglement and how the disentanglement is achieved.

$$\begin{aligned}\mathcal{L} = \mathbb{E}_{p(X)}\mathbb{E}_{q_\theta(X|Z_S,Z_C)}[-log(p_\theta(X|Z_S, Z_C))] + \\ \mathbb{E}_{p(X)}[\alpha kl(p(Z_S)||q_\theta(Z_S|X)) + \beta kl(p_\theta(Z_C)||q_\theta(Z_C|X))]\end{aligned} \tag{3}$$

where $kl(.,.)$ denotes KL divergence between two distributions, α and β are two balancing factors. The loss objective does not directly enforce the disentanglement between Z_S and Z_C, however, we will intuitively explain in the following sub-sections how the disentanglement could be achieved from three levels.

2.2.1. *Variational Mutual Information and KL Vanishing in VAE*

In the original form of VAE [15], the mutual information $MI(X, Z) = kl(p(X, Z)||p(X)p(Z))$ has its variational form $\hat{MI}_\theta(X, Z) = kl(q_\theta(X, Z)||p(X)p(Z)) = \mathbb{E}_{p(X)}[kl(q_\theta(Z|X)||p(Z))]$, which we call the variational mutual information. The objective of VAE can be reformulated as:

$$\mathcal{L}_{VAE}(\theta) = \mathbb{E}_{p(X)}\mathbb{E}_{q_\theta(Z|X)}[-log(p_\theta(X|Z))] + \hat{MI}_\theta(X, Z) \tag{4}$$

The first term in RHS of Eq. 4 is the reconstruction loss. It is observable that minimizing both two terms are typically not achievable considering that lower reconstruction loss means higher variational mutual information between X and Z, by assuming that we already achieve a good variational estimator, i.e. $q_\theta(Z|X) \approx p(Z|X)$. While there are exceptions when the decoder is designed to have powerful self-supervised generation capacity [18], these cases are not taken into consideration since the decoder is not such powerful in our scenario. Thus, VAE aims to reach a balance between the reconstruction loss and variational mutual information. It follows that the variational mutual information is lower bounded:

$$\inf_{\theta^*} \hat{MI}_{\theta^*}(X, Z) \geq A(X, Z) > 0 \tag{5}$$

where θ^* denotes a local optimal solution, $A(X, Z)$ is denotes the bounded variational mutual information, which is similar to I_C in Eq. 2 in [19]. The above formula guarantees that KL vanishing [18] will not happen during VAE training, which is the foundation of disentanglement.

2.2.2. *Information Flow Between Multiple Latent Variables in VAE*

We argue that balanced information flow between Z_S and Z_C leads to disentanglement. Based on the Eq. 4, the loss objective proposed in Eq. 3 can be formulated as:

$$\begin{aligned}\mathcal{L}(\theta) = \mathbb{E}_{p(X)}\mathbb{E}_{q_\theta(X|Z_S,Z_C)}[-log(p_\theta(X|Z_S, Z_C))] \\ + \alpha\hat{MI}_\theta(X, Z_S) + \beta\hat{MI}_\theta(X, Z_C)\end{aligned} \tag{6}$$

where the factorised form of [16] and [17] is a special case when $\alpha = \beta = 1$. Following Sec. 2.2.1, the summation of information encoded by Z_S and information encoded by Z_C is sufficient enough to reconstruct the input speech, that is:

$$\inf_{\theta^*}\{\alpha\hat{MI}_{\theta^*}(X, Z_S) + \beta\hat{MI}_{\theta^*}(X, Z_C)\} \geq A(X, Z) > 0 \tag{7}$$

Intuitively, by assuming convergence, when $\frac{\beta}{\alpha}$ is extremely large, the gradient of two variational mutual information terms will be dominated by $\hat{MI}(X, Z_C)$, which results in the vanishing of $KL(q_\theta(Z_C|X)||p(Z_C))$ so that the information encoded by the latent variable will be dominated by Z_S. Similar conclusion could be derived for the other side. The phenomenon of such imbalance is also observed in [20]. Theoretical expression of such gradient dominance is that, there exist $(\alpha_1, \beta_1), (\alpha_2, \beta_2)$ such that:

$$\alpha_1\hat{MI}_{\theta^*}(X, Z_S) + \beta_1\hat{MI}_{\theta^*}(X, Z_C) = \alpha_1\hat{MI}_{\theta^*}(X, Z_S) \tag{8}$$

$$\alpha_2\hat{MI}_{\theta^*}(X, Z_S) + \beta_2\hat{MI}_{\theta^*}(X, Z_C) = \beta_2\hat{MI}_{\theta^*}(X, Z_C) \tag{9}$$

We experimentally observe that the variational mutual information loss is independent of reconstruction loss, i.e. $\alpha_1\hat{MI}_{\theta^*}(X, Z_S) \approx \beta_2\hat{MI}_{\theta^*}(X, Z_C) \approx A(X, Z)$. In that sense, there is a double-sided information flow between Z_S and Z_C. By choosing a proper $\frac{\beta^*}{\alpha*}$, it would be possible that there is no overlap between the information encoded by Z_S and Z_C, i.e. $\hat{MI}_{\theta^*}(Z_S, Z_C) = 0$. Eq. 8 and Eq. 9 will be experimentally justified in Sec. 3.

2.2.3. Time-invariant and Time-Variant Disentanglement

We argue that applying average pooling techniques over the time dimension of a sequence of speech features leads to further disentanglement. According to Sec. 2.2.2, even if $\hat{MI}(Z_S, Z_C) = 0$, it is still undetermined which latent variable encodes speaker information or content information. Inspired by the 1D instancenorm method proposed in [14], we attempt to keep the speaker information by only taking the average pooling on speech features to derive the posterior distribution $q(Z_S|X)$. By choosing $\alpha = \alpha*$ and $\beta = \beta*$ as indicated in the second sufficient condition, Z_S will encode time-invariant information while Z_C will encode time-variant information. Other normalizations techniques like instancenorm2D also help a lot to obtain better disentanglement performance, which will be discussed in Sec. 3.

3. EXPERIMENTS

3.1. Dataset and Data Preprocessing

TIMIT We follow the official train/test split: 462 speakers for training and 24 speakers for testing [21]. For speaker verification (SV) task, we choose all possible trials from test set, which gives 18336 trials. 200 dimensional STFT features are extracted from a raw waveform with 25ms/10ms framing configuration. During training, the length of segment is fixed to 20 frames.

VCTK For VCTK corpus [22], 90% of the speakers are randomly selected for training and the remaining 10% as for testing. For the evaluation of disentanglement performance in the SV task, we generate 36900 trials (22950 nontarget trials and 14040 target trials) from test set. We extract melspectrogram as features with a framing configuration of 64ms/16ms. The feature dimension is set to 80. We select a segment of 100 frames for the VAE training.

3.2. Model Architectures

Encoder, Decoder, Vocoder There are two models designed for TIMIT and VCTK respectively. The encoder is composed of a shared encoder, speaker encoder and content encoder. (i) For TIMIT, the shared encoder is a 2-layer MLP with hidden size of 256. The content encoder is 2-layer BiLSTM with hidden size of 512, followed by a RNN layer with hidden size of 512. Then a 2-layer MLP of hidden size (512,64) is applied to model $q_\theta(Z_C|X)$. The speaker encoder is almost the same with content encoder except that there is an average pooling layer after RNN, and a 2-layer MLP is then applied to model $q_\theta(Z_S|X)$. The decoder is 1-layer MLP, followed by 2-layer BiLSTM which is followed by 2-layer MLP, where the hidden size is 256. Griffin-lim algorithm [23] is applied as vocoder. (ii) For VCTK, modified from [13], the shared encoder is composed of three convolutional layers with 512 channels. Each convolutional layer is followed by a linear layer with dimension 512 and an Instancenorm2D layer [24]. Also modified from [13], decoder includes a prenet with 512 channels and a postnet, which is a BiLSTM with hidden size of 512, followd by three convolutional layers with 512 channels, followed by a BiLSTM with hidden size of 512 and two separate linear layers to project the hidden dimension to 80 to model $p_\theta(X|Z_S, Z_C)$. A wavenet [25] pretrained on VCTK is used as Vocoder. Note that vocoder is only used for inference.

Prior Distribution The prior distribution includes $p(Z_S)$ and $p_\theta(Z_C)$. $p(Z_S)$ is modeled by a Normal distribution $N(0, \mathbb{I}_{d_S})$, where there are no parameters. $p_\theta(Z_C) = \prod_{t=1}^{T} P_\theta(Z_{Ct}|Z_{C\tau<t})$ is modeled by an auto-regressive LSTM with hidden size of 512. Each hidden cell is followed by two one-layer MLPs with hidden size 512 to model $P_\theta(Z_{Ct}|Z_{C\tau<t})$, from which Z_{Ct} is sampled and passed into the next LSTM cell.

3.3. Implementation Details

In the following, we first specify the hyper-parameters used for training, and then introduce evaluation metrics and training details. Next, the noise invariant method is introduced. Lastly, we detail the inference processes which include speaker verification, voice conversion.

Hyper-parameters Optimizer is adam with initial learning rate of 5e-4. Learning rate is decayed every 5 epochs with a factor of 0.95. Weight decay is 1e-4. Batchsize is 256. Both speaker dimension(d_S) and content dimension(d_C) are set to 64. We performed grid search for α and β and set $\alpha = 1, \beta = 20$ for TIMIT part and $\alpha = 0.01, \beta = 10$ for VCTK part. The embedding size $d_S = d_C = 64$.

Evaluation Metrics (i) EER (Equal Error Rate) is used as the evaluation metrics for disentanglement. Lower EER on speaker embeddings and higher EER on content embeddings typically (would be clarified in the discussion section) indicate better disentanglement, as observed in [16,17,26]. (ii) We conduct a mean opinion score test to evaluate our proposed approach. Two different set are provided for MOS test. For both seen to seen VC and unseen to unseen VC, we select 6 speakers (3 females and 3 males), each speaker with one utterance. So 30 test cases are included in the set. The listener needs to give a score for each sample in a test case according to the criterion: 1 = Bad; 2 = Poor; 3 = Fair; 4 = Good; 5 = Excellent. The final score for each model is calculated by averaging the collected results.

Noise Invariant training In the training of normal VAE, the VAE encoder takes clean acoustic feature X as the input, and produce embedding Z_S and Z_C. The decoder tries to reconstruct X with the feeding information Z_S and Z_C. To make the learned embedding Z_S and Z_C robust against background noise, we make a simple yet effective change in the training process. In the new data flow, a data augmentation module is applied to X, denoted as X^{aug}. The VAE encoder produces ${Z_S}^{aug}$ and ${Z_C}^{aug}$, which is the augmented version of Z_S and Z_C. The decoding process $D({Z_S}^{aug}, {Z_C}^{aug})$ reconstructs acoustic feature $\hat{X}^{aug}$, and we maximize the likelihood of $\hat{X}^{aug}$ and clean reference X. By introduce this inductive bias, we expect the VAE framework not only disentangle global and local information, but also perform denoising at the latent space. In this study, clean utterance is augmented by MUSAN dataset [27] with a balanced "noise", "music" and "babble" distribution.

Inference Experiments Since the inference tasks are performed on the utterance-level, we also start with making some notations. Denote an utterance $U = [X^{(1)}, X^{(2)}, ..., X^{(K)}]$, where $X^{(k)}$ is the kth segment. Denote ${Z_S}^{(k)}$ and ${Z_C}^{(k)}$ as the corresponding latent speaker and content variables for $X^{(k)}$. Here both ${Z_S}^{(k)}$ and ${Z_C}^{(k)}$ can be either the mean or a sample from $q({Z_S}^{(k)}|X^{(k)})$ and $q({Z_C}^{(k)}|X^{(k)})$ as we observed no significant difference made by these two. We implement two fundamental experiments: speaker verification and voice conversion.

(i) Speaker Verification. We apply the same method with [16, 17] to derive the speaker and content embedding: $\mu_S(U) = (\Sigma_{k=1}^{K} {Z_S}^{(k)})/K$ and $\mu_C(U) = (\Sigma_{k=1}^{K}(\Sigma_{t=1}^{T} {z_c}_t^{(k)}/T))/K$. For

TIMIT trials, we vary the value of $\frac{\beta}{\alpha}$ to observe the EER, and results are in Table 1. We implement t-SNE [28] visualization of speaker and content embeddings for VCTK, as shown in Fig. 2. For speaker embedding, the cluster pattern is clear, while for content embedding, the evenly distributed scatters demonstrates that only limited (if not zero) speaker information is leaked in the content embedding.

(ii) Voice Conversion. Denote U_{ij} as the converted utterance which takes U_i as source and U_j as target. We first obtain $\mu_S(U_j)$ as defined before. Second, pass $X_i^{(k)}$ into the model and we will have $Z_{C_i}^{(k)}$. in generation stage, $D(\mu_S(U_j), Z_{C_i}^{(k)})$ gives $\hat{X}_i^{(k)}$; then $U_{ij} = [\hat{X}_i^{(1)}, \hat{X}_i^{(2)}, ..., \hat{X}_i^{(K)}]$. We vary the value of $\frac{\beta}{\alpha}$ to observe the results of voice conversion. Since we do not observe big difference between VCTK and TIMIT parts, we just present the results for TIMIT in Fig. 3 to illustrate the double-sided information flow as proposed in the second sufficient condition.

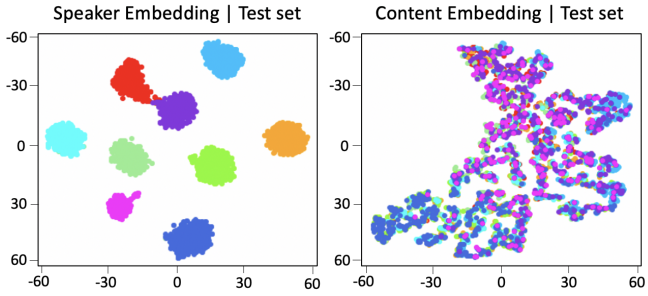

Fig. 2. Visualization of Speaker and Gender Clustering on VCTK

3.4. Results and discussions

Disentanglement As observed in Fig.3, only half of the utterance is converted to the target speaker for both two source utterances when $\frac{\beta}{\alpha} = 1$, which indicates that Z_S does not encode the whole speaker information and Z_C encodes part of the speaker information. When $\frac{\beta}{\alpha}$ is increased to 10 or 20, the speaker EER becomes lower and content EER becomes higher, which indicates better disentanglement. Fig.3 also illustrates this point since $\frac{\beta}{\alpha} = 10$ or 20 successfully achieves voice conversion. When $\frac{\beta}{\alpha} = 100$, Fig. 3 indicates that instead of performing voice conversion, it actually does utterance swapping, which means that Z_S encodes almost everything and Z_C encodes almost nothing. When this case happens, it is still reasonable that speaker EER is low and content EER is high, as observed in Table 1. To this end, we can derive another conclusion that good eer is actually the necessary but not sufficient condition for good disentanglement for two reasons. First, $\frac{\beta}{\alpha} = 1$ and $\frac{\beta}{\alpha} = 100$ give similar EER, however, there is no disentanglement for the latter case. Second, $\frac{\beta}{\alpha} = 10$ gives better EER, however, $\frac{\beta}{\alpha} = 20$ gives better disentanglement if we take a closer look at middle part of the spectrogram in male to female conversion in Fig. 3. The better way to look at disentanglement is to look at both EER and information flow. Fig.3 shows that as $\frac{\beta}{\alpha}$ is increasing, Z_S encodes more information and Z_C encodes less information, which is consistent with the assumption in our proposed second sufficient condition. For VCTK trails, the best SV EER of Z_S is 2.3%, while the EER of Z_S and Z_C w.r.t. the best VC performance are 4.6% and 44.5%. The difference indicates that balancing the information flow for disentanglement in VC remains a challenging problem.

Voice conversion[1] Table 2 shows the MOS results of different models. For fair comparison, we also generate VC samples from

[1]Samples of voice conversion can be found at https://jlian2.github.io/Robust-Voice-Style-Transfer

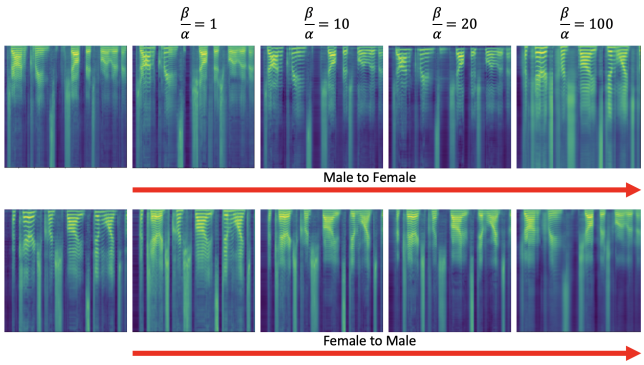

Fig. 3. Visualization of Double-sided Information Flow. Four models are trained with four different $\frac{\beta}{\alpha}$. Given a pair of male and female voice, we perform double-sided voice conversion. For each row, the first figure is the reconstructed spectrogram. The remaining four figures are the converted spectrograms. For the purpose of presentation, we just use fixed length of segment for voice conversion

Table 1. EER (%) for TIMIT test trials on varying $\frac{\beta}{\alpha}$.

$\frac{\beta}{\alpha}$	1	10	20	100	DSVAE [16]
μ_S	5.40	3.25	4.16	5.01	4.94
μ_C	31.09	38.83	37.16	38.79	17.49

AUTOVC and AdaIN-VC, where we use their pretrained models with our generated VC pairs. As illustrated in the table, our proposed system outperforms AUTOVC and AdaIN-VC with a large margin in both naturalness and similarity. To test the effectiveness of noise invariant training, we conduct an additional MOS test. For test the noise invariant VC model, we apply background noise with signal to noise ratio range of 3-10 dB. As indicated in the MOS test, we only see marginal performance degradation compared with clean input, which shows the effectiveness of noise invariant training.

Table 2. The results of the MOS (95% CI) test on different models.

model	seen to seen		unseen to unseen	
	naturalness	similarity	naturalness	similarity
AUTOVC [13]	2.65±0.12	2.86±0.09	2.47±0.10	2.76±0.08
AdaIN-VC [14]	2.98±0.09	3.06±0.07	2.72±0.11	2.96±0.09
Ours	3.40±0.07	3.56±0.06	3.22±0.09	3.54±0.07
Ours(noisy)	3.23±0.09	3.43±0.07	3.12±0.08	3.47±0.08

4. CONCLUSION

In this study, we proposed a novel zero-shot voice conversion system. By decomposing speech into speaker representation Z_C and content representation Z_S with a sequential VAE framework, the conversion can be as simple as: a) swapping the speaker representation; b) feeding source content embedding and target speaker embedding to the decoder to generate the acoustic features; c) convert the acoustic feature to the waveform with a vocoder. We improved the DSVAE framework by analyzing the information flow global between speaker representation and local content representation. Noise invariant training was investigated, which enabled the VC system to handle low quality speech input. For both SV and VC tasks, we achieved the state-of-the-art system performance on TIMIT and VCTK. We believe this study serves as a preliminary research work, and can be beneficial to other domains, such as speech recognition, text to speech studies.

5. REFERENCES

[1] Berrak Sisman, Junichi Yamagishi, Simon King, and Haizhou Li, "An overview of voice conversion and its challenges: From statistical modeling to deep learning," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2020.

[2] Fahimeh Bahmaninezhad, Chunlei Zhang, and John HL Hansen, "Convolutional neural network based speaker de-identification.," in *ISCA Odyssey*, 2018.

[3] Liqiang Zhang, Chengzhu Yu, Heng Lu, Chao Weng, Chunlei Zhang, Yusong Wu, Xiang Xie, Zijin Li, and Dong Yu, "Durian-sc: Duration informed attention network based singing voice conversion system," *arXiv preprint arXiv:2008.03009*, 2020.

[4] Yannis Stylianou, Olivier Cappé, and Eric Moulines, "Continuous probabilistic transform for voice conversion," *IEEE Transactions on speech and audio processing*, vol. 6, no. 2, pp. 131–142, 1998.

[5] Tomoki Toda, Alan W Black, and Keiichi Tokuda, "Voice conversion based on maximum-likelihood estimation of spectral parameter trajectory," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 8, pp. 2222–2235, 2007.

[6] Donald J Berndt and James Clifford, "Using dynamic time warping to find patterns in time series," in *KDD workshop*, 1994, pp. 359–370.

[7] Jing-Xuan Zhang, Zhen-Hua Ling, Li-Juan Liu, Yuan Jiang, and Li-Rong Dai, "Sequence-to-sequence acoustic modeling for voice conversion," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 3, pp. 631–644, 2019.

[8] Hirokazu Kameoka, Takuhiro Kaneko, Kou Tanaka, and Nobukatsu Hojo, "Stargan-vc: Non-parallel many-to-many voice conversion using star generative adversarial networks," in *2018 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2018, pp. 266–273.

[9] Takuhiro Kaneko and Hirokazu Kameoka, "Cyclegan-vc: Non-parallel voice conversion using cycle-consistent adversarial networks," in *2018 26th European Signal Processing Conference (EUSIPCO)*. IEEE, 2018, pp. 2100–2104.

[10] Lifa Sun, Kun Li, Hao Wang, Shiyin Kang, and Helen Meng, "Phonetic posteriorgrams for many-to-one voice conversion without parallel data training," in *2016 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2016, pp. 1–6.

[11] Haohan Guo, Heng Lu, Na Hu, Chunlei Zhang, Shan Yang, Lei Xie, Dan Su, and Dong Yu, "Phonetic posteriorgrams based many-to-many singing voice conversion via adversarial training," *arXiv preprint arXiv:2012.01837*, 2020.

[12] Mingyang Zhang, Yi Zhou, Li Zhao, and Haizhou Li, "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.

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

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

[15] Diederik P Kingma and Max Welling, "Auto-encoding variational bayes," 2014.

[16] Yingzhen Li and Stephan Mandt, "Disentangled sequential autoencoder," *arXiv preprint arXiv:1803.02991*, 2018.

[17] Yizhe Zhu, Martin Renqiang Min, Asim Kadav, and Hans Peter Graf, "S3vae: Self-supervised sequential vae for representation disentanglement and data generation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 6538–6547.

[18] Junxian He, Daniel Spokoyny, Graham Neubig, and Taylor Berg-Kirkpatrick, "Lagging inference networks and posterior collapse in variational autoencoders," *arXiv preprint arXiv:1901.05534*, 2019.

[19] Alexander A Alemi, Ian Fischer, Joshua V Dillon, and Kevin Murphy, "Deep variational information bottleneck," *arXiv preprint arXiv:1612.00410*, 2016.

[20] Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn, "Gradient surgery for multi-task learning," *arXiv preprint arXiv:2001.06782*, 2020.

[21] John S Garofolo, "Timit acoustic phonetic continuous speech corpus," *Linguistic Data Consortium, 1993*, 1993.

[22] Christophe Veaux, Junichi Yamagishi, and Kirsten MacDonald, "Cstr vctk corpus: English multi-speaker corpus for cstr voice cloning toolkit," 2017.

[23] 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.

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

[25] 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.

[26] Wei-Ning Hsu, Yu Zhang, and James Glass, "Unsupervised learning of disentangled and interpretable representations from sequential data," *arXiv preprint arXiv:1709.07902*, 2017.

[27] David Snyder, Guoguo Chen, and Daniel Povey, "Musan: A music, speech, and noise corpus," *arXiv preprint arXiv:1510.08484*, 2015.

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