INTERSPEECH 2023
20-24 August 2023, Dublin, Ireland

S2CD: Self-heuristic Speaker Content Disentanglement for Any-to-Any Voice Conversion

Pengfei Wei[1], *Xiang Yin*[1], *Chunfeng Wang*[1], *Zhonghao Li*[1], *Xinghua Qu*[1], *Zhiqiang Xu*[2], *Zejun Ma*[1]

[1]Bytedance, Singapore
[2]Machine Learning Department, Mohamed bin Zayed University of Artificial Intelligence, Abu Dhabi, UAE

[1]{pengfei.wei, yinxiang.stephen, wangchunfeng, lizhonghao.01, xinghua.qu, mazejun}@.bytedance.com, [2]zhiqiang.xu@mbzuai.ac.ae

Abstract

In this paper, we propose a **S**elf-heuristic **S**peaker **C**ontent **D**isentanglement (S2CD) model for *any_to_any* voice conversion without using any external resources, e.g., speaker labels or vectors, linguistic models, and transcriptions. S2CD is built on the disentanglement sequential variational autoencoder (DSVAE), but improves DSVAE structure at the model architecture level from three perspectives. Specifically, we develop different structures for speaker and content encoders based on their underlying static/dynamic property. We further propose a generative graph, modelled by S2CD, so as to make S2CD well mimic the multi-speaker speech generation process. Finally, we propose a self-heuristic way to introduce bias to the prior modelling. Extensive empirical evaluations show the effectiveness of S2CD for *any_to_any* voice conversion.

Index Terms: voice conversion, any_to_any, disentanglement

1. Introduction

Voice conversion (VC) aims to convert the timbre of a speech from a source speaker to a target speaker while preserving the content of the source speech. Based on the use of training data, VC models can be categorized into parallel and non-parallel ones, where the former is technically simpler but less practical while the latter is more challenging and attractive. Precisely, non-parallel VC can be further divided following a "*src_to_tgt*" naming convention, where *src* and *tgt* belong to {*one*, *many*, *any*}. *One* represents an extreme case where a single speaker is used either in training or inference. In contrast, *many* and *any* involve multiple speakers, and the difference is that speakers are seen in *many* but unseen in *any*, during the training.

In the last decade, numerous methods have been proposed for non-parallel VC. Although these methods generally share the same idea of learning speaker and content representations, the technical details (e.g., whether using automatic speech recognition (ASR) guided techniques [1, 2, 3], or text-to-speech (TTS) guided techniques [4, 5, 6], or mixed techniques [7, 8, 9]) and external dependencies (whether using speaker labels/vectors [10, 11, 12], transcriptions [13, 14, 15], or other auxiliary models [16, 17, 18]) differ considerably under different "*src_to_tgt*" VC scenarios.

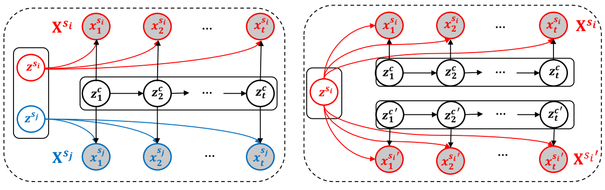
Phonetic posterior-grams and pre-trained speaker models were widely used for the content and speaker representations in *any_to_one* VC where the speech corpus of a target speaker is fixed, e.g., [19, 20]. Recently, discretized self-supervised speech representations are proposed to boost *any_to_one* VC. For instance, Huang et al. [8] propose to use VQW2V [21] to eliminate speaker information and represent speech by discrete speech units. SoftVC [7] achieves further improvements by using Hubert [22] to extract soft speech units. Moreover, there are also methods utilizing off-the-shelf TTS techniques to extract speaker-independent linguistic features, e.g., Cotatron [13] takes advantage of Tacotron2 [23] and Mix-Guided VC [9] combines ASR and TTS encoders. However, all these models usually require sufficient training data with rich transcriptions and speaker labels, e.g., [9, 13], and even a large amount of external well-annotated data, e.g., [7, 8, 19], which are very expensive to collect.

Many_to_many VC is a more flexible setting that converts voice among speakers within a speaker training set. Mainstream related research contains generative adversarial network (GAN) based [24, 25, 26, 27] and (variational) autoencoder ((V)AE) based [16, 28, 29, 30]. The key idea of GAN-based VC methods is to learn speaker-indistinguishable representation through adversarial learning. An early work CycleGAN-VC [24] utilizes adversarial and the cycle-consistency losses, but is limited to one-to-one mapping. StarGan-VC [25] improves to many-to-many mapping by adding another domain classification loss. CC-GAN [26] further uses a speaker-conditional encoder and a multi-output discriminator to simplify the model structure and boost the VC performance. Recently, Ma et al. [27] develop an SGAN-VC using subband block to perform style transfer for each frequency. Note that GAN-based VC methods usually require speaker labels/vectors to train the discriminator. Moreover, the learning objective of these methods usually consists of several losses, e.g., up to 7 losses in [27]. Balancing such many losses is challenging, and the generalization is thus limited.

AE/VAE-based method is another popular research line for *many_to_many* VC. Basically, these methods aim to disentangle the speaker and content information from speech data. The very first work [28] simply uses a conventional VAE to disentangle the content embedding, and then incorporate it with a pretrained speaker vector for VC. A later work ACVAE-VC [16] uses a speaker-conditioned content posterior and introduces an auxiliary classifier for speaker prediction. Instead of conditioning on speaker attributes, [29] exploits a pitch tracker to construct an F0-conditioned AE. Note that all these methods need an auxiliary speaker model outside VAE/AE structure. Recently, Luong et al. [30] propose a disentanglement VAE based on a directed graph that directly models speaker and content latent factors, avoiding the need for external speaker vectors.

Compared with *Many_to_many* VC, *any_to_any* VC is a more general scenario, where VC happens between any speakers even they are unseen during training. Due to its generalizability, *any_to_any* VC research becomes increasingly popular. Pioneering works, AutoVC [10] and AdaIN-VC [31], follow an AE structure and use information bottleneck to separate speaker and content information. A later work VQVC+ [32] uses vector quantization (VQ) to extract discrete linguistic representations and eliminate the speaker information. However, these meth-

10.21437/Interspeech.2023-215



(a) Same utterance from different speakers

(b) Different utterances from same speaker

Figure 1: *Graphic models of S2CD-VC.*

ods still reply on the pretrained speaker models. VQMIVC [33] and IDE-VC [34] improve VQVC+ and AutoVC, repsectively, by explicitly building speaker encoder and adding a mutual information (MI) loss, while AGAIN-VC [35] improves AdaIN-VC through a unified encoder and an activation to guide the training. Although technically sound, VQMIVC and IDE-VC suffers from the complex training process due to the difficulty in estimating MI, and AGAIN-VC is not robust to balance the quality of speech audio and the similarity of speaker style.

Except for pretrained speaker models, some *any_to_any* VC works also take advantage of other external models. Both [36] and [37] adopt wav2vec [38] to extract linguistic embedding, while [39] utilizes speaker verification model [17] to facilitate the speaker modelling. The availability and quality of external models play an important role in the VC success of these methods. GAN-based ideas are also exploited in *any_to_any* VC, e.g., [11, 12, 40]. However, same as stated in *many_to_many* VC, these methods usually stack a large amount of losses, specifically 5 in [40, 11] and 7 in [12], and thus lack of generalizability. There are also methods achieving VC by TTS-based system, e.g., YourTTS [6] and STYLETTS [15], with rich transcription or phonemes available for training.

Very recently, VAE-based methods have shown a great success in *any_to_any* VC. In [41], the authors propose a variant of β-VAE [42] that is specifically for disentanglement of content and speaker representations by two individual latent factors. A later work [43] further investigates a more powerful VAE model, disentangled sequential VAE (DSVAE) [44], specifically disentangling time-invariant and time-variant information from sequential data. Such VAE-based methods are elegant for *any_to_any* VC in the sense that they fully rely on the strong disentanglement capacity of model itself without using any external resources (e.g., pretrained linguistic models, transcriptions and speaker labels/vectors) and auxiliary losses (only reconstruction and KL losses are used in the learning objective). A very latest work CDSVAE [14] further shows that the VC performance can be boosted using external models or transcriptions to introduce content bias to the prior modeling.

In this paper, we aim to propose a novel VC method without using any external resources for *any_to_any* VC. We build on DSVAE that explicitly models speaker and content latent factors. However, different from [14, 43] that directly adopt the original structure of DSVAE [44], we develop more advanced submodule structures at the model architecture level to better serve for VC purpose, meanwhile, without adding extra losses to vanilla DSVAE objectives. Specifically, we propose the following three improvements.

Firstly, we design different structures for content and speaker encoders, rather than to use the same one, i.e., BiLSTM, as [14, 43, 44] do. Considering that content and speaker latent factors encode dynamic and static information, respectively, we propose to use BiLSTM and transformer without positional embedding as the base model of content and speaker encoders, correspondingly. We intuitively and empirically show that such design benefits the disentanglement[1].

Secondly, to further enhance the benefit of disentanglement to VC, we enforce our model to follow a generative graph as shown in Fig.1. Ideally, identical utterance of different speakers is generated from the same content latent factor but different speaker latent factors, while different utterances of the same speaker should share the same speaker latent factor. As parallel data is absent, we put more focus on the latter. Specifically, we propose to feed positive pairs of utterances (utterances from the same speaker) into speaker encoder to model a shared speaker latent factor by using an average function.

Thirdly, we also introduce related bias to the prior modelling. However, instead of using external speaker or linguistic models, we propose a self-heuristic way. We build the prior directly using speaker and content representations sampled from the corresponding posteriors. With these improvements, we obtain our **S**elf-heuristic **S**peaker **C**ontent **D**isentanglement model (S2CD) for *any_to_any* VC. To summarize, the main contribution of this paper is as follows.

- We present a comprehensive review of existing non-parallel VC methods, discussing their technical details and external dependencies under different “*src_to_tgt*” VC scenarios.
- We propose an S2CD model for *any_to_any* VC without using any external resources. S2CD-VC is based on DSVAE, but improves DSVAE from three perspectives.
- We conduct extensive empirical evaluations, including comparison with existing methods and property analyses, on S2CD to show its effectiveness for *any_to_any* VC.

2. The Proposed Method

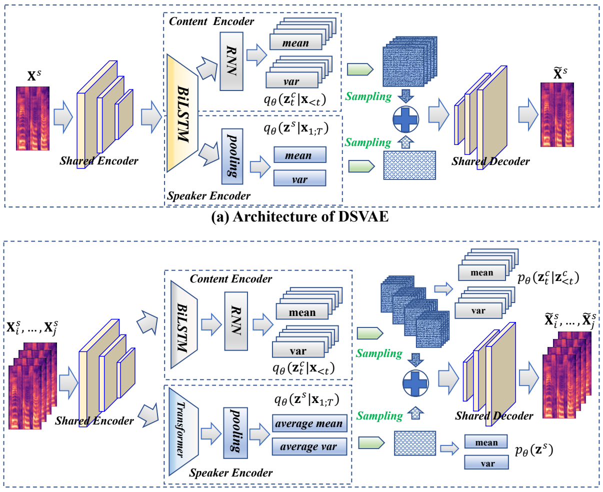
2.1. Preliminary

We start with the problem formulation of *any_to_any* VC. Let $\mathbf{X} = [\mathbf{x}_1, ..., \mathbf{x}_T]$ be a T-segment speech, represented by acoustic features, e.g., melspectrogram, of a speaker sampled from $\mathcal{S} = [s_1, ..., s_n]$. Our goal is to train a model with the speech data of multiple speakers from $\mathcal{S}$ without using any external resources, for *any_to_any* VC. The test includes two scenarios, namely *seen2seen*: $s_i \rightarrow s_j$ where $s_i, s_j \in \mathcal{S}$ and *unseen2unseen*: $s_i \rightarrow s_j$ where $s_i, s_j \notin \mathcal{S}$.

As our S2CD uses DSVAE as the backbone, we introduce DSVAE herein. The latest DSVAE architecture adopted in [14] is shown in Fig.2(a). The input melspectrograms are fed into a shared encoder, which consists of several convolutional layers, to extract the deep acoustic features, followed by a BiLSTM to explore the temporal information. Two groups of *mean* and *variance* networks are applied to model the posterior of speaker and content latent factors, *i.e.*, $q_\theta(\mathbf{z}^s|\mathbf{X})$ and $q_\theta(\mathbf{z}_t^c|\mathbf{x}_{<t})$. The new representations $\mathbf{z}_1^c, ..., \mathbf{z}_T^c$ and $\mathbf{z}^s$ are correspondingly sampled. Each $\mathbf{z}_t^c$ is then concatenated with $\mathbf{z}^s$ and passed to a decoder for reconstruction. Finally, the reconstructed melspectrograms are gone through a vocoder to construct the waveform. The prior of $\mathbf{z}^s$ and $\mathbf{z}_{1:t}^c$ is a standard Gaussian distribution and modelled by an autoregressive LSTM, respectively. Both posterior and prior distributions follow the conditional independence assumption same as [44]. The overall learning objective then consists of reconstruction and KL-divergence parts:

$$\mathcal{L} = \mathcal{L}_{rec} + \lambda_s \mathcal{L}_{kld_s} + \lambda_c \mathcal{L}_{kld_c}, \tag{1}$$

[1] Uing different structures for encoder is a general improvement for DSVAE [44] beyond VC task, as it essentially aims to strength the disentanglement of static and dynamic information.


$\mathbf{X}^s$
Shared Encoder
Content Encoder
BiLSTM
RNN
mean
var
$q_\theta(\mathbf{z}_t^c|\mathbf{x}_{<t})$
$q_\theta(\mathbf{z}^s|\mathbf{x}_{1:T})$
pooling
Speaker Encoder
Sampling
Sampling
Shared Decoder
$\tilde{\mathbf{X}}^s$
$\mathbf{X}_i^s, \ldots, \mathbf{X}_j^s$
Transformer
average mean
average var
$p_\theta(\mathbf{z}_t^c|\mathbf{z}_{<t}^c)$
$p_\theta(\mathbf{z}^s)$
$\tilde{\mathbf{X}}_i^s, \ldots, \tilde{\mathbf{X}}_j^s$

(a) Architecture of DSVAE

(b) Improved architecture of S2CD

Figure 2: *S2CD and DSVAE architecture comparison.*

where

$$\begin{aligned}\mathcal{L}_{rec} &= \mathbb{E}_{p(\mathbf{X})}\mathbb{E}_{q_\theta(\mathbf{X}|\mathbf{z}^s,\mathbf{z}_{1:T}^c)}[-\log(q_\theta(\mathbf{X}|\mathbf{z}^s,\mathbf{z}_{1:T}^c))],\\ \mathcal{L}_{kld_s} &= \mathbb{E}_{p(\mathbf{X})}[KL(q_\theta(\mathbf{z}^s)|\mathbf{X})||p_\theta(\mathbf{z}^s)],\\ \mathcal{L}_{kld_c} &= \mathbb{E}_{p(\mathbf{X})}[KL(q_\theta(\mathbf{z}_t^c|\mathbf{z}_{<t}^c)||p_\theta(\mathbf{z}_t^c|\mathbf{z}_{<t}^c))].\end{aligned}$$

In the test phase, given $\mathbf{X}_{src}$ and $\mathbf{X}_{tgt}$, we first feed them into the well-trained DSVAE, and then concatenate each $\mathbf{z}_{src}^c$ with $\mathbf{z}_{tgt}^s$. The converted melspectrogram can be obtained by passing the above cancatenated vectors to the decoder. We then use a vocoder to convert the melspectrogram to the waveform.

2.2. The proposed S2CD

DSVAE is an elegant model for *any_to_any* VC as it does not require any external resources or introduce any other constraints, during training. It succeeds in VC thanks to its strong capability of disentangling the static and dynamic information from speech data. In this paper, we further improve DSVAE at the model architecture level so that we can preserve the above elegance of DSVAE for VC. We call the improved model **S**elf-heuristic **S**peaker **C**ontent **D**isentanglement (S2CD) for VC. The underlying intuition of the proposed improvements is to strengthen the consistency of the static-dynamic disentanglement and speaker-content embedding modelling. In other words, we further enhance the disentanglement capability of the model on the one hand, and on the other hand ensure that static and dynamic latent factors indeed encode the speaker and content information, respectively. Next, we introduce the three improvements we have proposed in details.

2.2.1. Customized Encoder Structure

As shown in Fig.2(a), in DSVAE, speaker and content encoders share a BiLSTM module, followed by an RNN to construct temporal latent factors for the content encoder and a pooling module to extract a global latent factor for the speaker encoder. This structure is reasonable but could result in the temporal information leaking to the speaker encoder due to the shared BiLSTM. To avoid so, we propose to use customized structures for the two encoders. Precisely, we keep the structure of the content encoder, i.e., BiLSTM+RNN, but utilize a transformer encoder without positional embedding for the speaker encoder, i.e., Transformer+Pooling. By doing so, we can structurally make the speaker encoder avoid using temporal information.

Looking deeper, if we randomly shuffle the temporal order of frames to form a shuffled sequence, the static factors of the original and shuffled sequences should be ideally equal or at least close. BiLSTM+Pooling fails to make this happen, but Transformer+Pooling qualifies as it is permutation-invariant [45]. Essentially, such customized encoder structure enables better disentanglement of static and dynamic information.

2.2.2. Positive Pair-wise Training

Our ultimate goal is not just to achieve a better disentanglement, but to obtain a better disentanglement that benefits the VC task. Thus, our second improvement focuses on the alignment of static-dynamic disentanglement and speaker-content embedding modelling. To do so, we propose a generative graph for the generation of multi-speaker speech as shown in Fig.1, and enforce our model to follow the generative graph. Fig.1 shows two ideal cases, one modelling the generation of the same utterance from different speakers and another modelling that of different utterances from the same speaker. As we focus on non-parallel VC, we use the latter to guide the disentanglement.

Specifically, during training, we propose to feed pair-wise utterances from the same speaker into the speaker encoder. We then use an average function over the obtained two speaker vectors to construct the shared one. Afterwards, the shared speaker latent factor will be used for reconstruction for both utterances. By doing so, we can explicitly ensure that the pair-wise utterances are generated from the same speaker latent factor. Herein, we only use pair-wise utterances, but the number of positive utterances can be extended to more. Moreover, instead of simply using the average function, more sophisticated functions can be investigated. These are potential points worthy further study.

2.2.3. Self-heuristic Prior Modelling

CDSVAE [14] shows that involving content bias in the prior modeling improves DSVAE that uses standard priors. However, it requires auxiliary resources, e.g., transcriptions, to obtain the content prior knowledge. Thus, our third improvement tries to model a good prior without using external resources. We propose a self-heuristic way, that is, we use the sampled speaker and content representations to build their respective priors. The intuition behind is that a good disentanglement leads to a good sampled new representation containing speaker/content information rich enough to construct the corresponding priors. We note that such a self-heuristic way may have a weaker prior modelling compared with CDSVAE, but the shining point is its independence on any external resources.

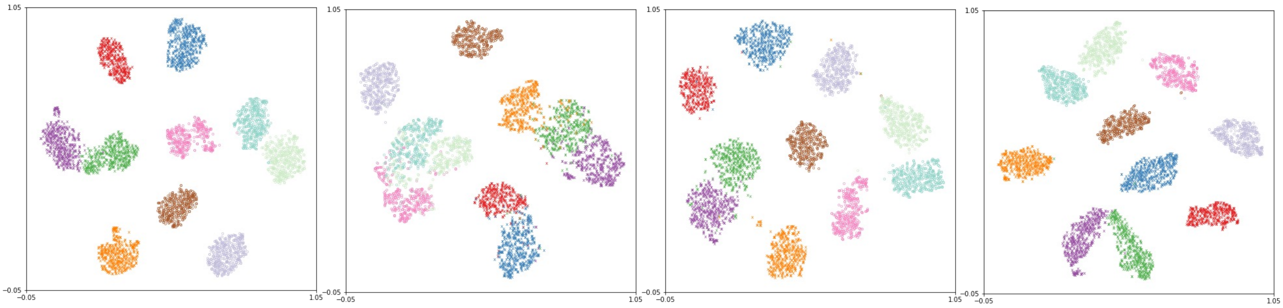
2.2.4. Overall Architecture

In summary, we propose three improvements over DSVAE in the model architecture level. Fig.1(b) shows the architecture of our proposed S2CD, and the difference from DSVAE can be clearly observed compared with Fig.1(a). All the mean and variance networks are dense layers, and the decoder consists of a prenet and a postnet following [10]. We use Melgan [46] as vocoder due to its fast inference speed. The vocoder is pretrained and not fine-tuned in S2CD training.

3. Experiments

3.1. Experimental Configuration

We use VCTK dataset [47] for experimental study. Following [14], we randomly select 10 speakers forming unseen speaker



(a) DSVAE (b) $\text{S2CD}_{w/o\_T\&P}$ (c) $\text{S2CD}_{w/o\_T}$ (d) S2CD

Figure 3: *Visualizations of speaker latent embedding in the unseen2unseen scenario.*

Table 1: *The MOS test with 95% CI.*

Methods	Seen2Seen		Unseen2Unseen	
	Naturalness	*Similarity*	*Naturalness*	*Similarity*
AutoVC	1.37 ±0.06	2.85 ±0.24	1.58 ±0.21	3.46 ±0.18
DSVAE_{ours}	3.18 ±0.09	2.93 ±0.18	3.38 ±0.10	3.15 ±0.11
CDSVAE_{ours}	3.84 ±0.10	3.31 ±0.18	3.69 ±0.10	3.27 ±0.10
DSVAE [14]	3.76 ±0.07	3.83 ±0.06	3.65 ±0.07	3.89 ±0.05
CDSVAE [14]	4.03 ±0.04	4.12 ±0.07	3.93 ±0.06	4.06 ±0.07
$\text{S2CD}_{w/o\_T\&P}$	4.05 ±0.09	3.91 ±0.13	3.99 ±0.08	3.82 ±0.11
$\text{S2CD}_{w/o\_T}$	4.11 ±0.06	4.02 ±0.09	4.04 ±0.08	3.92 ±0.08
S2CD	4.32 ±0.04	4.21 ±0.07	4.22 ±0.06	4.02 ±0.07

Table 2: *Phoneme classification accuracy (%).*

Methods	Mel_only	DSVAE	$\text{S2CD}_{w/o\_T\&P}$	$\text{S2CD}_{w/o\_T}$	S2CD
Accuracy	59.75	47.2	63.95	64.47	64.77

set, while the rest speakers are used for training. We extract melspectrogram as features with a framing configuration of 64ms/16ms, and set feature dimension to 80. We select a segment of 64 frames for training. Adam is used as the optimizer with fixed learning rate of 1e-4. The dimension of the convolutional layers in shared encoder is 256, and that of speaker and content latent factors is set to 64. Batch size is set to 256. For λ_s and λ_c, we use 0.1 and 1, respectively, according to the grid search. All the experiments are done on NVIDIA A100 GPU.

3.2. Subjective Evaluation

We first evaluate S2CD by the mean opinion score (MOS) test. For both *seen2seen* and *unseen2unseen* scenarios, we randomly select 30 test cases. Each test case includes two utterances from a source speaker and a target speaker randomly selected from the corresponding speaker set, as well as the converted utterance. Ten listeners evaluate the converted utterances by giving scores from 1 to 5 on naturalness and similarity. The final score is calculated by averaging all the collected results. The demo samples are shown in this link[2].

We compare with several baselines including AutoVC [10], DSVAE and CDSVAE [14]. As the code of [14] is not public, we implement it according to the paper. Unfortunately, we fail to reproduce the scores[3]. Thus we show the results of our run and also the paper-reported numbers. For S2CD, we introduce two more variants, namely $\text{S2CD}_{w/o\_T\&P}$ and $\text{S2CD}_{w/o\_T}$. We gradually add the three improvements to DSVAE, firstly the pair-wise training, resulting in $\text{S2CD}_{w/o\_T\&P}$, followed by self-heuristic prior modelling, resulting in $\text{S2CD}_{w/o\_T}$, and lastly customized encoders, giving us the final S2CD.

The MOS results are shown in Fig.1. As can be seen, our reproduced DSVAE and CDSVAE results are generally lower than the paper-reported ones, but the trend is the same, i.e., CDSVAE improves DSVAE. We mainly compare with the reported scores. For our methods, $\text{S2CD}_{w/o\_T\&P}$ achieves comparable and better results compared with DSVAE on similarity and naturalness, respectively, but is still not as good as CDSVAE, especially on similarity. By adding self-heuristic prior modelling, $\text{S2CD}_{w/o\_T}$ generally catches up with CDSVAE with better naturalness but weaker similarity. Further taken customized encoders into account, S2CD finally outperforms CDSVAE in average. Note that S2CD has another superiority over CDSVAE, i.e., its independence on external resources. The performance gain from $\text{S2CD}_{w/o\_T\&P}$ to S2CD also shows the effectiveness of each improvement for VC.

3.3. Latent Factors Analyses

We also analyze the disentanglement performance by (1) visualizing speaker latent embeddings and (2) performing phoneme classification with content latent embeddings. We show the t-SNE visualization and phoneme classification results on the test speaker set in Fig.3 and Table.2, respectively. DSVAE obtains a clear cluster pattern in Fig.3(a). However, the worst phoneme classification accuracy of DSVAE, 47.3% even worse than mel_only, is also observed. This shows the room of further improvements on disentanglement. For $\text{S2CD}_{w/o\_T\&P}$, it has looser clusters but better phoneme classification performance than DSVAE, which overall balances the performance gain. This is consistent with their comparable MOS in Table.1. $\text{S2CD}_{w/o\_T}$ achieves similar distributed clusters but clear better accuracy than DSVAE. Thus, we observe the performance gain of $\text{S2CD}_{w/o\_T}$ over DSVAE on MOS. For the final S2CD, it achieves not only denser clusters but also higher accuracy compared with all the other baselines, and thus is the best on VC performance. This set of experiments show that our proposed improvements indeed lead to a better disentanglement for VC.

4. Conclusions

In this paper, we first present a comprehensive review of existing non-parallel VC methods under different "*src_to_tgt*" scenarios. We then focus on the latest "*any_to_any*" model DSVAE, and propose an S2CD model with three improvements, i.e., customized encoder structures, positive pair-wise training, and slef-heuristic prior modelling, over DSVAE. Empirical results show S2CD is a promising method for "*any_to_any*" VC.

[2] https://wmaiga.github.io/S2CD/

[3] The model structure is well stated in [14], and we suspect the non-reproducibility is due to some imperceivable differences of our implementation from the official one.

5. References

[1] J. Zhang, Z. Ling, L. Liu, Y. Jiang, and L. Dai, "Sequence-to-sequence acoustic modeling for voice conversion," *Trans. Audio, Speech and Lang. Proc.*, pp. 631–644, 2019.

[2] X. Zhao, F. Liu, C. Song *et al.*, "Disentangling content and fine-grained prosody information via hybrid asr bottleneck features for voice conversion," in *ICASSP*, 2022, pp. 7022–7026.

[3] Y. Chen, L. Liu, Y. Hu, Y. Jiang, and Z. Ling, "Improving recognition-synthesis based any-to-one voice conversion with cyclic training," in *ICASSP*, 2022, pp. 7007–7011.

[4] M. Zhang, Y. Zhou, L. Zhao, and H. Li, "Transfer learning from speech synthesis to vc with non-parallel training data," *Trans. Audio, Speech and Lang. Proc.*, pp. 1290–1302, 2021.

[5] K. Kim, S. Park, J. Lee, and M. Joe, "Assem-vc: Realistic voice conversion by assembling modern speech synthesis techniques," in *ICASSP*, 2022, pp. 6997–7001.

[6] E. Casanova, J. Weber, C. D. Shulby, A. C. Junior, E. Gölge, and M. A. Ponti, "Yourtts: Towards zero-shot multi-speaker tts and zero-shot vc for everyone," in *ICML*, 2022, pp. 2709–2720.

[7] B. van Niekerk, M.-A. Carbonneau, J. Zaïdi *et al.*, "A comparison of discrete and soft speech units for improved voice conversion," in *ICASSP*, 2022, pp. 6562–6566.

[8] W. Huang, Y. Wu, and T. Hayashi, "Any-to-one sequence-to-sequence voice conversion using self-supervised discrete speech representations," in *ICASSP*, 2021, pp. 5944–5948.

[9] Z. Zhao, S. Ma, Y. Jia, J. Hou, L. Yang, and J. Wang, "Mix-guided vc: Any-to-many voice conversion by combining asr and tts bottleneck features," in *ISCSLP*, 2022, pp. 96–100.

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

[11] Q. Wang, X. Zhang, J. Wang *et al.*, "Drvc: A framework of any-to-any vc with self-supervised learning," in *ICASSP*, 2022.

[12] B. Nguyen and F. Cardinaux, "Nvc-net: End-to-end adversarial voice conversion," in *ICASSP*, 2022, pp. 7012–7016.

[13] S. Park, D. Kim, and M. Joe, "Cotatron: Transcription-guided speech encoder for any-to-many voice conversion without parallel data," in *Interspeech*, 2020, p. 4696–4700.

[14] J. Lian, C. Zhang, G. K. Anumanchipalli, and D. Yu, "Towards improved zero-shot voice conversion with conditional dsvae," in *Interspeech*, 2022, p. 2598–2602.

[15] Y. A. Li, C. Han, and N. Mesgarani, "Styletts-vc: One-shot voice conversion by knowledge transfer from style-based tts models," in *SLT workshop*, 2023, pp. 920–927.

[16] H. Kameoka, T. Kaneko, K. Tanaka, and N. Hojo, "Acvae-vc: Non-parallel vc with auxiliary classifier vae," *Trans. Audio, Speech and Lang. Proc.*, 2019.

[17] A. Mittal and M. Dua, "Automatic speaker verification systems and spoof detection techniques: review and analysis," *International Journal of Speech Technology*.

[18] T. Merritt, A. Ezzerg, P. Biliński *et al.*, "Text-free non-parallel many-to-many vc using normalising flow," in *ICASSP*, 2022.

[19] L. Sun, K. Li, H. Wang *et al.*, "Phonetic posteriorgrams for many-to-one vc without parallel data training," in *ICME*, 2016.

[20] S. Liu, J. Zhong, L. Sun, X. Wu, X. Liu, and H. Meng, "Voice conversion across arbitrary speakers based on a single target-speaker utterance." in *Interspeech*, 2018, pp. 496–500.

[21] A. Baevski, S. Schn., and M. Auli, "vq-wav2vec: Self-supervised learning of discrete speech representations," in *ICLR*, 2020.

[22] W. Hsu, B. Bolte, Y. H. Tsai *et al.*, "Hubert: Self-supervised speech representation learning by masked prediction of hidden units," *Trans. Audio, Speech and Lang. Proc.*, 2021.

[23] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang *et al.*, "Natural tts synthesis by conditioning wavenet on mel spectrogram predictions," in *ICASSP*, 2018, pp. 4779–4783.

[24] T. Kaneko and H. Kameoka, "Cyclegan-vc: Non-parallel voice conversion using cycle-consistent adversarial networks," in *EUSIPCO*, 2018, pp. 2100–2104.

[25] H. Kameoka, T. Kaneko, K. Tanaka, and N. Hojo, "Stargan-vc: Non-parallel many-to-many voice conversion using star generative adversarial networks," in *SLT*, 2018, pp. 266–273.

[26] S. Lee, B. Ko, K. Lee, I.-C. Yoo, and D. Yook, "Many-to-many voice conversion using conditional cycle-consistent adversarial networks," in *ICASSP*, 2020, pp. 6279–6283.

[27] J. Ma, Z. Zheng, H. Fei, F. Zheng, T. Chua, and Y. Yang, "Subband-based gan for non-parallel many-to-many vc," *arXiv preprint arXiv:2207.06057*, 2022.

[28] C.-C. Hsu, H.-T. Hwang, Y.-C. Wu, Y. Tsao, and H.-M. Wang, "Voice conversion from non-parallel corpora using variational auto-encoder," in *APSIPA*, 2016.

[29] K. Qian, Z. Jin, M. Hasegawa J, and G. J. Mysore, "F0-consistent many-to-many non-parallel voice conversion via conditional autoencoder," in *ICASSP*, 2020, pp. 6284–6288.

[30] M. Luong and V. A. Tran, "Many-to-many vc based feature disentanglement using vae," in *Interspeech*, 2021, pp. 851–854.

[31] J. Chou, C. Yeh, and H. Lee, "One-shot voice conversion by separating speaker and content representations with instance normalization," in *Interspeech*, 2019, pp. 664–668.

[32] D.-Y. Wu, Y.-H. Chen, and H.-Y. Lee, "Vqvc+: One-shot voice conversion by vector quantization and u-net architecture," in *Interspeech*, 2020, pp. 4691–4695.

[33] D. Wang, L. Deng, Y. Yeung *et al.*, "Vqmivc: Vector quantization and mutual information-based unsupervised speech representation disentanglement for one-shot vc," in *Interspeech*, 2021.

[34] S. Yuan, P. Cheng, R. Zhang, W. Hao, Z. Gan, and L. Carin, "Improving zero-shot voice style transfer via disentangled representation learning," in *ICLR*, 2021.

[35] Y. Chen, D. Wu, T. Wu, and H. Lee, "Again-vc: A one-shot voice conversion using activation guidance and adaptive instance normalization," in *ICASSP*, 2021, pp. 5954–5958.

[36] Y. Y. Lin, C. Chien, J. Lin, H. Lee, and L. Lee, "Fragmentvc: Any-to-any vc by end-to-end extracting and fusing fine-grained voice fragments with attention," in *ICASSP*, 2021, pp. 5939–5943.

[37] T. Dang, D. Tran, P. Chin, and K. Koishida, "Training robust zero-shot voice conversion models with self-supervised features," in *ICASSP*, 2022, pp. 6557–6561.

[38] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," *NeurIPs*, pp. 12 449–12 460, 2020.

[39] R. Xiao, H. Zhang, and Y. Lin, "Dgc-vector: A new speaker embedding for zero-shot vc," in *ICASSP*, 2022, pp. 6547–6551.

[40] W. Li and T.-J. Wei, "Asgan-vc: One-shot voice conversion with additional style embedding and generative adversarial networks," in *APSIPA ASC*, 2022, pp. 1932–1937.

[41] H. Lu, D. Wang, X. Wu, Z. Wu, X. Liu, and H. Meng, "Disentangled speech representation learning for one-shot cross-lingual vc using ß-vae," in *SLT Workshop*, 2022, pp. 814–821.

[42] I. Higgins, L. Matthey, A. Pal, C. Burgess *et al.*, "beta-vae: Learning basic visual concepts with a constrained variational framework," in *ICLR*, 2017.

[43] J. Lian, C. Zhang, and D. Yu, "Robust disentangled variational speech representation learning for zero-shot voice conversion," in *ICASSP*, 2022, pp. 6572–6576.

[44] L. Yingzhen and S. Mandt, "Disentangled sequential autoencoder," in *ICML*, 2018, pp. 5670–5679.

[45] A. Vaswani, N. Shazeer, N. Parmar *et al.*, "Attention is all you need," *NuerIps*, 2017.

[46] K. Kumar, R. Kumar, T. De Boissiere, L. Gestin, W. Z. Teoh, J. Sotelo, A. de Brébisson, Y. Bengio, and A. C. Courville, "Melgan: Generative adversarial networks for conditional waveform synthesis," *NuerIps*, 2019.

[47] V. Christophe, Y. Junichi, and M. Kirsten, "Cstr vctk corpus: English multi-speaker corpus for cstr voice cloning toolkit," 2017.