Interspeech 2024
1-5 September 2024, Kos, Greece

PRVAE-VC2: Non-Parallel Voice Conversion by Distillation of Speech Representations

Kou Tanaka, Hirokazu Kameoka, Takuhiro Kaneko, Yuto Kondo

NTT Corporation, Japan

kouef.tanaka@ntt.com

Abstract

This paper describes a knowledge distillation approach to non-parallel many-to-many voice conversion (VC) using self-supervised speech representation techniques: perturbation-resistant variational autoencoder (PRVAE) and a hidden-unit BERT (HuBERT). PRVAE has achieved a breakthrough by significantly improving non-streaming and low-latency streaming VC performances. However, a notable gap persists between the real target and converted speech, posing a continuing challenge. To narrow this gap, we present PRVAE-VC2, an improved version of PRVAE-VC that leverages rich and contextually informed representations derived from pre-trained HuBERT. Furthermore, we apply knowledge distillation techniques to make PRVAE-VC2 a streamable method, ensuring that the advantage of PRVAE-VC as a streamable VC is not compromised. Evaluation results demonstrate that our approaches effectively reduce discrepancies between the converted speech and the target. Audio samples can be accessed at our webpage [1].

Index Terms: Voice conversion, speech representation, knowledge distillation, variational autoencoder, self-supervised learning

1. Introduction

Voice conversion (VC) is a technique that transforms the speech of one speaker to sound like that of another while preserving linguistic content. This technique finds applications in various domains, including speaker conversion, assistive systems for overcoming speech and hearing impairments, and pronunciation and accent conversions for language learning. In this paper, we specifically focus on speaker timbre conversion.

Three frameworks for learning conversion models exist parallel VC, ASR-TTS VC, and non-parallel VC. Parallel VC [1–4] requires a parallel speech corpus consisting of recordings of the same sentences spoken by both source and target speakers. Collecting such a corpus is time-consuming and expensive, but it can be optimized directly based on the target speech, potentially yielding high-quality results. ASR-TTS VC [5, 6], as the name implies, is an approach cascading automatic speech recognition (ASR) and text-to-speech synthesis (TTS). It utilizes a phoneme recognizer to extract phonetic information from the input speech, which is then fed to TTS to generate the output speech. This method has shown the best performance in the Voice Conversion Challenge [7], but it requires text data paired with speech to train ASR and TTS, which is a limiting factor. Non-parallel VC [8–14], on the other hand, converts source speech to target speech without relying on any paired data, such as text aligned with speech or target speech aligned with source speech. This makes the task more challenging, as the model has to learn the correspondence between the source and target speech without any guidance from parallel data. However, non-parallel VC has become an active research area in recent years due to the availability of many non-parallel speech data.

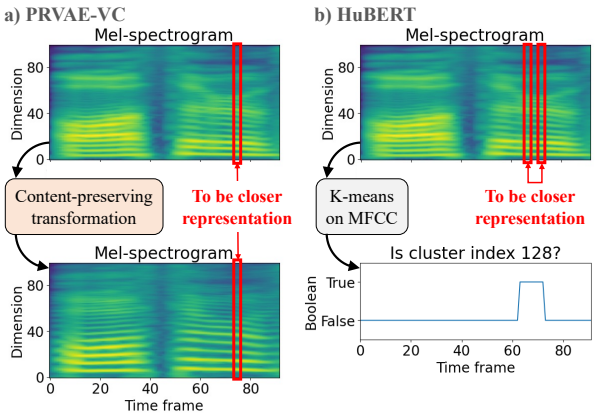


Figure 1: *An overview of semantic information compression. The sentence is 'Auther.' Both methods reduce the dimensionality of the input space along different axes. Therefore, our proposal incorporates both methods in an additive manner.*

Early attempts [8–12] at non-parallel VC employ techniques such as autoencoders (AE) [15], variational autoencoders (VAE), and generative adversarial networks (GAN) [16]. These methods assume that finding better intermediate features enables the model to generalize more effectively to unseen data and different speakers. Among them, perturbation-resistant VAE-based VC (PRVAE-VC) [17] is a powerful method for learning less speaker-dependent speech representations. It achieves this by imposing a simple constraint that aligns the latent features of perturbed-speech with those of the original speech, as shown in Fig. 1(a). The perturbed-speech is generated through content-preserving transformations of the original speech's fundamental frequency and spectral envelope, simulating the manipulation of speaker identity. Thanks to this straightforward constraint, the conversion performance has been improved for non-streaming VC and low-latency streaming VC. However, a notable gap persists between the real target and converted speech.

Latter attempts [13, 14] at non-parallel VC involve hierarchical and unsupervised models for learning speech representations, such as contrastive predictive coding (CPC) [18], wav2vec [19, 20], hidden-unit BERT (HuBERT) [21], and

[1] http://www.kecl.ntt.co.jp/people/tanaka.ko/projects/prvaevc2/

10.21437/Interspeech.2024-947

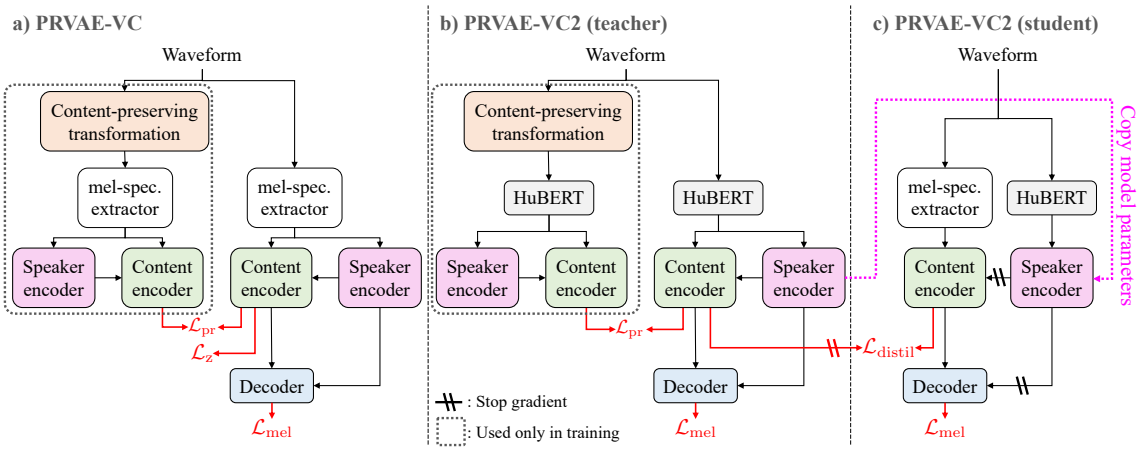


Figure 2: *A system overview of the conventional PRVAE-VC and the proposed PRVAE-VC2. Black and red solid arrows indicate the conversion flow and the loss calculation, respectively.*

WavLM [22]. These speech representation learning methods aim to capture rich, contextual, and meaningful features from speech signals without the need for manual annotations or labeled training data. By pre-training on large amounts of unlabeled audio data, these models learn to extract domain-agnostic features that are robust and generalizable across various downstream tasks, including ASR, TTS, and VC. The key idea behind these methods is that discretization creates an information bottleneck that separates content and speaker details. By mapping similar continuous values to the same discrete symbols, discretization helps compress semantic information in the input space, as shown in Fig. 1(b). This compression forces the model to capture the most salient features of the audio signal while discarding redundant or less informative details. As a result, the representations learned by the model become more compact and semantically meaningful.

Inspired by speech representation learning methods, PRVAE-VC [17] and HuBERT [21], we propose PRVAE-VC2, which improves upon PRVAE-VC by leveraging rich, contextually informed representations from pre-trained HuBERT. PRVAE-VC and HuBERT obtain meaningful speech representation by reducing the dimensionality of the input space on different axes. Therefore, we expect both methods of compressing semantic information in the input space to be additively effective. Furthermore, as HuBERT is not a lightweight model, we apply knowledge distillation techniques to make PRVAE-VC2 a streamable method, ensuring that the advantages of PRVAE-VC as a streamable VC method are not compromised. Our experimental results demonstrated that introducing perturbation resistance for the output of the pre-trained HuBERT successfully improved the speaker similarity score, sound quality score, and character error rate of the converted speech. Moreover, the results of the knowledge distillation to make the proposed method streamable also showed that the performance of PRVAE-VC2 significantly surpassed that of PRVAE-VC.

Our main contributions are as follows:

- The constraints that were effective in PRVAE-VC and the discretization that was effective in HuBERT are shown to be effective in an additive manner.
- We find that better performance is obtained when the loss term $\mathcal{L}_{\mathrm{z}}$ is not considered, as shown in Fig. 2(b). (Namely, the latent variable does not follow a standard Gaussian distribution in PRVAE-VC2.)
- We show that our knowledge distillation approach is efficient for high-fidelity streaming VC.

2. Conventional PRVAE-VC

A perturbation-resistant (PR) variant [17] of VAE [23], called PRVAE, is a neural network model that includes speaker encoder, content encoder, and decoder networks, as shown in Fig. 2(a). The speaker encoder takes the mel-spectrogram $\boldsymbol{x}$ as input and produces a speaker vector $\boldsymbol{c}$ as output. Then, to obtain the reconstructed mel-spectrogram $\hat{\boldsymbol{x}}$, the content encoder and decoder networks are designed to generate the sequences of the means and logarithmic variances, $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$, as follows:

$$\boldsymbol{c} = \mathrm{SpeakerEncoder}(\boldsymbol{x}), \tag{1}$$

$$\left[\boldsymbol{\mu}_{\boldsymbol{z}}; \log \boldsymbol{\sigma}_{\boldsymbol{z}}^2\right] = \mathrm{ContentEncoder}(\boldsymbol{x}, \boldsymbol{c}), \tag{2}$$

$$\left[\boldsymbol{\mu}_{\hat{\boldsymbol{x}}}; \log \boldsymbol{\sigma}_{\hat{\boldsymbol{x}}}^2\right] = \mathrm{Decoder}(\boldsymbol{\mu}_{\boldsymbol{z}} + \boldsymbol{\sigma}_{\boldsymbol{z}} \odot \boldsymbol{\epsilon}, \boldsymbol{c}), \tag{3}$$

$$\hat{\boldsymbol{x}} = \boldsymbol{\mu}_{\hat{\boldsymbol{x}}} + \boldsymbol{\sigma}_{\hat{\boldsymbol{x}}} \odot \boldsymbol{\epsilon}, \tag{4}$$

where $\boldsymbol{z}$, $\boldsymbol{\epsilon}$, $[;]$, and $\odot$ denote a latent space variable, Gaussian noise, concatenation along the channel dimension, and element-wise multiplication.

One unique aspect of PRVAE-VC is its ability to train the latent variable representations of the original speech and the speech altered by content-preserving transformation to be as consistent as possible. To obtain parameters for the distribution of latent space variable $\boldsymbol{z}_m$ of content-preserving transformed speech, the mel-spectrogram $\boldsymbol{x}_m$ extracted from the speech performed F_0 shift and frequency warping is fed to the speaker encoder and content encoder, as follows:

$$\boldsymbol{c}_m = \mathrm{SpeakerEncoder}(\boldsymbol{x}_m), \tag{5}$$

$$\left[\boldsymbol{\mu}_{\boldsymbol{z}_m}; \log \boldsymbol{\sigma}_{\boldsymbol{z}_m}^2\right] = \mathrm{ContentEncoder}(\boldsymbol{x}_m, \boldsymbol{c}_m). \tag{6}$$

To train the content encoder to match the two distributions of the latent space variables $\boldsymbol{z}$ and $\boldsymbol{z}_m$, PR loss is defined as follows:

$$\mathcal{L}_{\mathrm{pr}} = \lambda_{\mathrm{pr}} F_{\mathrm{KLD}}\left(\mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{z}}, \boldsymbol{\sigma}_{\boldsymbol{z}}^2) | \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{z}_m}, \boldsymbol{\sigma}_{\boldsymbol{z}_m}^2)\right), \tag{7}$$

where $\mathcal{N}(\cdot)$, F_{KLD}, and λ_{pr} denote a Gaussian distribution, a KL divergence loss function, and a regularization parameter controlling the importance of the PR regularization, respectively.

The full objective function $\mathcal{L}$ of PRVAE-VC is given as,

$$\mathcal{L} = \mathcal{L}_{\mathrm{z}} + \mathcal{L}_{\mathrm{mel}} + \mathcal{L}_{\mathrm{pr}}, \tag{8}$$

$$\mathcal{L}_{\mathrm{z}} = F_{\mathrm{KLD}}\big(\mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{z}}, \boldsymbol{\sigma}^2_{\boldsymbol{z}}) | \mathcal{N}(\mathbf{0}, \boldsymbol{I})\big), \tag{9}$$

$$\mathcal{L}_{\mathrm{mel}} = F_{\mathrm{GNLL}}\big(\boldsymbol{x}, \mathcal{N}(\boldsymbol{\mu}_{\hat{\boldsymbol{x}}}, \log \boldsymbol{\sigma}^2_{\hat{\boldsymbol{x}}})\big), \tag{10}$$

where F_{GNLL} denotes a Gaussian negative log-likelihood loss function.

3. PRVAE-VC2

The proposed method's model is learned in a two-stage learning process, which is a teacher-student model training, as shown in Fig. 2(b) and 2(c). The teacher model uses the pre-trained HuBERT model instead of the mel-spectrogram extractor for more accurate modeling. We train a streamable VC model as a student model by applying knowledge distillation techniques to the trained teacher model.

3.1. Stage1: Teacher model training with HuBERT and PR regularization

As mentioned in Sec. 1, both HuBERT and PRVAE-VC are speech representation learning methods. However, these two methods reduce the dimensionality of the input space along different axes by using discretization and PR regularization, respectively. Based on the above, we expect the discretization and PR regularization to be additively effective, and we use both techniques to train the teacher model.

The model architectures of the teacher model are unchanged from those of the conventional method PRVAE-VC, except that HuBERT features are used as input for the speaker encoder and content encoder instead of mel-spectrograms. On the other hand, for the objective function $\mathcal{L}_{\mathrm{teacher}}$, the loss term $\mathcal{L}_{\mathrm{z}}$ is excluded, as follows:

$$\mathcal{L}_{\mathrm{teacher}} = \mathcal{L}_{\mathrm{mel}} + \mathcal{L}_{\mathrm{pr}}. \tag{11}$$

Note that Sec. 4.3.1 will describe that the loss term $\mathcal{L}_{\mathrm{z}}$ is unnecessary when the discretization and PR regularization are used.

3.2. Stage2: Student model training by using knowledge distillation techniques

While the conventional PRVAE-VC consists only of streamable architectures and achieves low-latency streaming VC, the proposed teacher model is unsuitable for low-latency streaming VC because HuBERT is generally not a lightweight model. In order to use PRVAE-VC2 as a low-latency streaming VC in the same way as conventional PRVAE-VC, knowledge distillation techniques are applied to well-trained teacher models to train streamable student models.

When the low-latency streaming VC is used, the speaker vectors of both source and target speakers are usually precalculated. Therefore, the speaker encoder's model size does not adversely affect the operating speed of the low-latency streaming VC. To ensure the operational feasibility of PRVAE-VC2, the architecture of the student model should be the same as the conventional PRVAE-VC, except for the speaker encoder.

We train student models under different conditions and evaluate their performance to find the best-performing student model that can extract knowledge from the trained teacher model. Although the results of the ablation study will be described in Sec. 4.3.2, the model shown in Fig. 2(c) achieved the best performance and was trained with the objective function $\mathcal{L}_{\mathrm{student}}$ as follows:

$$\mathcal{L}_{\mathrm{student}} = \mathcal{L}_{\mathrm{mel}} + \mathcal{L}_{\mathrm{distil}}, \tag{12}$$

$$\mathcal{L}_{\mathrm{distil}} = F_{\mathrm{KLD}}\big(\mathcal{N}_{\mathrm{t}}(\boldsymbol{\mu}_{\boldsymbol{z}}, \boldsymbol{\sigma}^2_{\boldsymbol{z}}) | \mathcal{N}_{\mathrm{s}}(\boldsymbol{\mu}_{\boldsymbol{z}}, \boldsymbol{\sigma}^2_{\boldsymbol{z}})\big), \tag{13}$$

where $\mathcal{N}_{\mathrm{t}}$ and $\mathcal{N}_{\mathrm{s}}$ denote the Gaussian distributions with the distribution parameters derived from the teacher and student models, respectively. Using the loss term $\mathcal{L}_{\mathrm{distil}}$, the effects of discretization by HuBERT and regularization by PRVAE are carried over to the student model. Note that the speaker encoder is copied from the teacher model and fixed during the training.

The remaining conditions not used in the training of the best-performing student model but used in the ablation study are defined as follows: (p1) a method of learning the speaker encoder of the student model rather than copying the speaker encoder of the teacher model and fixing it during the training, (p2) a method that uses the decoder of the teacher model as that of the student model and fixes it during the training, (p3) mel-spectrograms converted by the teacher model to have the voice timbre of other speakers are used as pseudo-target paired with source speech.

4. Experiments

4.1. Implementation details

The architectures of the speaker encoder, content encoder, and decoder were the same as those of the large model reported in [17], except that the number of dimensions of the input of encoders was 80 for mel-spectrograms and 768 for HuBERT features. mel-spectrograms were extracted from 16 kHz speech with a hop length of 10 ms and a Han window of 64 ms. On the other hand, the hop length of pre-trained HuBERT was 20 ms because we used HuBERT-Base model reported in [21]. Therefore, we achieved a hop length of 10 ms by alternately placing HuBERT features extracted from 10 ms shifted speech and HuBERT features from the original. Any neural vocoders conditioned on mel-spectrograms as input can be used to generate the waveform from the given mel-spectrogram. In the experiment, iSTFTNet [24] was used as the neural vocoder. We set a regularization parameter λ_{pr} to 1.0. Other model settings were the same as those reported in [17].

4.2. Other experimental conditions

We conducted experimental evaluations using the CMU ARCTIC database [25], which contains parallel recordings of professional US English speakers. To train VC models, we used 1000 sentences per speaker. To evaluate the performance, we used 132 sentences per speaker. All models were trained on *many-to-many* condition, which is 4-speaker input and 4-speaker output. Therefore, the number of speech samples evaluated is 1584 (132 sentences $\times$ 12 speaker pairs) on each VC system.

As the evaluation metrics, we used character error rate (CER) [%], speaker similarity score (SIM), and 5-scaled pseudo mean opinion score (MOS) on sound quality. The CER was calculated by the joint CTC-attention Transformer-based ASR model [26] trained on LibriSpeech [27], provided by ESPnet [28]. Given the converted speech and the reference speech of the target speaker, the SIM was calculated by the ECAPA-TDNN model [29] trained on VoxCeleb [30, 31], provided by SpeechBrain [32]. The MOS was calculated by the official UTMOS-strong model [33] trained on the Voice-MOS Challenge 2022 dataset [34].

Table 1: *Evaluation results. The lower the CER, the better the performance. The higher the SIM and MOS, the better the performance. The values following ± indicate confidence intervals. "AnaSyn" and "voc." indicate the analysis-by-synthesized speech and vocoder, respectively. The method* **B1**, **T1**, *and* **S1** *are shown in Fig. 2. The SIM of* **X1** *and* **X2** *were calculated using speech pairs of the same speaker but different sentences. Note that the SIM was 0.087 ± 0.006 for the same sentence with different speakers.*

No.	Method	CER ↓	SIM ↑	MOS ↑
X1	Recorded	1.31	$0.722_{\pm 0.007}$	$4.39_{\pm 0.01}$
X2	AnaSyn	1.36	$0.717_{\pm 0.007}$	$4.22_{\pm 0.02}$
Baseline				
B1	PRVAE-VC	6.80	$0.661_{\pm 0.004}$	$3.63_{\pm 0.02}$
B2	w/o $\mathcal{L}_{\mathrm{pr}}$	10.92	$0.625_{\pm 0.005}$	$3.37_{\pm 0.02}$
B3	w/o $\mathcal{L}_{\mathrm{z}}$	1.52	$0.266_{\pm 0.007}$	$3.53_{\pm 0.02}$
B4	w/o $\mathcal{L}_{\mathrm{z}}$ and $\mathcal{L}_{\mathrm{pr}}$	1.46	$0.118_{\pm 0.005}$	$2.94_{\pm 0.04}$
BF	w/ finetuned voc.	6.79	$0.703_{\pm 0.004}$	$3.91_{\pm 0.01}$
Proposed (teacher)				
T1	PRVAE-VC2	1.40	$0.700_{\pm 0.004}$	$3.87_{\pm 0.01}$
T2	w/ $\mathcal{L}_{\mathrm{z}}$ and w/o $\mathcal{L}_{\mathrm{pr}}$	2.37	$0.703_{\pm 0.004}$	$3.71_{\pm 0.01}$
T3	w/ $\mathcal{L}_{\mathrm{z}}$	2.14	$0.700_{\pm 0.004}$	$3.69_{\pm 0.02}$
T4	w/o $\mathcal{L}_{\mathrm{pr}}$	1.44	$0.614_{\pm 0.005}$	$3.80_{\pm 0.02}$
TF	w/ finetuned voc.	1.41	$0.741_{\pm 0.004}$	$4.22_{\pm 0.01}$
Proposed (student with teacher as T1)				
S1	PRVAE-VC2	1.59	$0.691_{\pm 0.004}$	$3.96_{\pm 0.01}$
S2	w/ p1 in Sec. 3.2	1.70	$0.694_{\pm 0.004}$	$3.97_{\pm 0.01}$
S3	w/ p2 in Sec. 3.2	1.63	$0.682_{\pm 0.004}$	$3.73_{\pm 0.02}$
S4	w/ p3 in Sec. 3.2	1.70	$0.683_{\pm 0.004}$	$3.75_{\pm 0.02}$
SF	w/ finetuned voc.	1.58	$0.718_{\pm 0.004}$	$4.21_{\pm 0.01}$

4.3. Results

4.3.1. Effect of using HuBERT and PR regularization

To evaluate the effectiveness of the PRVAE-VC2 proposed in Sec. 3, we comprehensively compared the use/non-use of loss terms for the model training. The evaluation results are shown in Table 1. First, a comparison between the conventional model **B1** and the proposed teacher model **T3** shows that when learning from scratch, the use of HuBERT features performs better than the use of mel-spectrogram in terms of CER, SIM, and MOS. Moreover, comparing teacher models **T3** and **T2** shows that PR regularization is effective even when using HuBERT features. Therefore, we found that the discretization by HuBERT and regularization by PRVAE are additively effective.

Next, by comparing the teacher models **T1** and **T3** of the proposed method, we confirmed that the model **T1** trained without the loss term $\mathcal{L}_{\mathrm{z}}$ achieved the better performance. This means that when using HuBERT and PR regularization, it is better to eliminate the constraint that the latent variables follow the standard normal distribution commonly employed in VAE.

4.3.2. Effect of knowledge distillation approaches

The effectiveness of using HuBERT and PR regularization in an additive manner was demonstrated in the previous section. Here, we will confirm that the generalization capabilities of **T1** can be distilled to student models which have the same model architecture as the conventional approach. Table 1 shows that, Similar to the results of the teacher model, the proposed student model **S1** outperformed the conventional method **B1** in all of CER, SIM, and MOS. This indicates that the knowledge distillation works well. The student model uses a content encoder with mel-spectrogram as input, similar to the conventional method, but thanks to the knowledge distillation loss $\mathcal{L}_{\mathrm{distil}}$, it shows that it is able to learn a better speech representation than that of the conventional method.

Next, the student models **S1**, **S2**, **S3**, and **S4** of the proposed method are compared to be clear which distillation method is preferred. The result that the CER of **S2** is worse than that of **S1** indicates that it is better not to train the speaker encoder of the student model but to use the speaker encoder of the teacher model as that of the student model. As shown in Fig. 3, the speaker encoder of **S1** may estimate more generalized speaker representations than those of **B1** and **S2**. Furthermore, the comparison of **S1**, **S3**, and **S4** shows that it is better to constrain only the intermediate features than to constrain the decoder side that generates the mel-spectrogram.

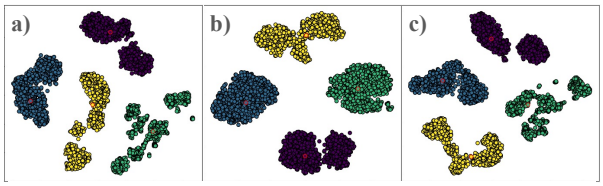


Figure 3: *Visualization of speaker vectors on the training data. a), b) and c) are the results on the system* **B1**, **S1**, *and* **S2**, *respectively. Note that b) is also the result on* **T1**, **S3**, *and* **S4** *because of the same speaker encoder. Each color indicates each speaker. The points circled in red indicate the centroid of each speaker.*

4.3.3. Effect of vocoder finetuning

To investigate the final results, the vocoder finetuning technique was applied to **B1** as the conventional method, **T1** as the teacher model of the proposed method, and **S1** as the student model. The reconstructed mel-spectrograms via each method and the corresponding speech waveform are used as the paired data to train the finetuned vocoder.

Comparing the CER, SIM, and MOS of the methods using finetuned vocoder (**BF**, **TF**, and **SF**) with those of the methods using conventional vocoders (**B1**, **T1**, and **S1**), we found that SIM and MOS are significantly improved by vocoder finetuning. In particular, the SIM and MOS of the converted speech on the proposed methods **TF** and **SF** are comparable to those of the analysis-by-synthesis speech denoted as **X2**.

5. Conclusions

We proposed a knowledge distillation approach for non-parallel many-to-many VC integrating PRVAE and HuBERT, which are self-supervised speech representation techniques. As the results, we showed that 1) the constraints that were effective in PRVAE-VC and the discretization that was effective in HuBERT are shown to be effective in an additive manner, and 2) the knowledge distillation technique makes it possible to dramatically improve performance even if using the same size content encoder and decoder as the conventional PRVAE-VC.

6. References

[1] J. Zhang, Z. Ling, L. Liu, Y. Jiang, and L. 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.

[2] K. Tanaka, H. Kameoka, T. Kaneko, and N. Hojo, "AttS2S-VC: Sequence-to-sequence voice conversion with attention and context preservation mechanisms," in *ICASSP*, pp. 6805–6809, 2019.

[3] H. Kameoka, K. Tanaka, D. Kwaśny, T. Kaneko, and N. Hojo, "ConvS2S-VC: Fully convolutional sequence-to-sequence voice conversion," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 1849–1863, 2020.

[4] H. Kameoka, W.-C. Huang, K. Tanaka, T. Kaneko, N. Hojo, and T. Toda, "Many-to-many voice transformer network," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 656–670, 2020.

[5] J.-X. Zhang, Z.-H. Ling, and L.-R. Dai, "Non-parallel sequence-to-sequence voice conversion with disentangled linguistic and speaker representations," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 540–552, 2019.

[6] L.-J. Liu, Y.-N. Chen, J.-X. Zhang, Y. Jiang, Y.-J. Hu, Z.-H. Ling, and L.-R. Dai, "Non-parallel voice conversion with autoregressive conversion model and duration adjustment," in *Joint Workshop for the Blizzard Challenge and Voice Conversion Challenge*, pp. 126–130, 2020.

[7] Z. Yi, W.-C. Huang, X. Tian, J. Yamagishi, R. K. Das, T. Kinnunen, Z.-H. Ling, and T. Toda, "Voice Conversion Challenge 2020 — Intra-lingual semi-parallel and cross-lingual voice conversion —," in *Joint Workshop for the Blizzard Challenge and Voice Conversion Challenge*, pp. 80–98, 2020.

[8] T. Kaneko and H. Kameoka, "CycleGAN-VC: Parallel-Data-Free Voice Conversion Using Cycle-Consistent Adversarial Networks," in *EUSIPCO*, pp. 2100–2104, 2018.

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

[10] 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*, pp. 266–273, 2018.

[11] H. Kameoka, T. Kaneko, K. Tanaka, and N. Hojo, "ACVAE-VC: Non-parallel voice conversion with auxiliary classifier variational autoencoder," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 9, pp. 1432–1443, 2019.

[12] T. Kaneko, H. Kameoka, K. Tanaka, and N. Hojo, "MaskCycleGAN-VC: Learning non-parallel voice conversion with filling in frames," in *ICASSP*, pp. 5919–5923, 2021.

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

[14] W.-C. Huang, S.-W. Yang, T. Hayashi, H.-Y. Lee, S. Watanabe, and T. Toda, "S3PRL-VC: Open-source voice conversion framework with self-supervised speech representations," in *ICASSP*, pp. 6552–6556, 2022.

[15] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *science*, vol. 313, no. 5786, pp. 504–507, 2006.

[16] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *NeurIPS*, pp. 2672–2680, 2014.

[17] K. Tanaka, H. Kameoka, and T. Kaneko, "PRVAE-VC: Non-Parallel Many-to-Many Voice Conversion with Perturbation-Resistant Variational Autoencoder," in *ISCA Speech Synthesis Workshop*, 2023.

[18] A. van den Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *arXiv preprint arXiv:1807.03748*, 2018.

[19] S. Schneider, A. Baevski, R. Collobert, and M. Auli, "wav2vec: Unsupervised Pre-Training for Speech Recognition," in *INTERSPEECH*, pp. 3465–3469, 2019.

[20] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," *Advances in neural information processing systems*, vol. 33, pp. 12 449–12 460, 2020.

[21] W.-N. Hsu, B. Bolte, Y.-H. H. Tsai, K. Lakhotia, R. Salakhutdinov, and A. Mohamed, "HuBERT: Self-Supervised Speech Representation Learning by Masked Prediction of Hidden Units," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 3451–3460, 2021.

[22] S. Chen, C. Wang, Z. Chen, Y. Wu, S. Liu, Z. Chen, J. Li, N. Kanda, T. Yoshioka, X. Xiao *et al.*, "WavLM: Large-Scale Self-Supervised Pre-Training for Full Stack Speech Processing," *IEEE Journal of Selected Topics in Signal Processing*, vol. 16, no. 6, pp. 1505–1518, 2022.

[23] D. P. Kingma and M. Welling, "Auto-Encoding Variational Bayes," in *ICLR*, 2014.

[24] T. Kaneko, K. Tanaka, H. Kameoka, and S. Seki, "iSTFTNet: Fast and lightweight mel-spectrogram vocoder incorporating inverse short-time fourier transform," in *ICASSP*, pp. 6207–6211, 2022.

[25] J. Kominek and A. W. Black, "The CMU Arctic speech databases," in *ISCA Speech Synthesis Workshop*, pp. 223–224, 2004.

[26] S. Karita, N. Chen, T. Hayashi, T. Hori, H. Inaguma, Z. Jiang, M. Someki, N. E. Y. Soplin, R. Yamamoto, X. Wang *et al.*, "A comparative study on transformer vs rnn in speech applications," in *ASRU*, pp. 449–456, 2019.

[27] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *ICASSP*, pp. 5206–5210, 2015.

[28] S. Watanabe, T. Hori, S. Karita, T. Hayashi, J. Nishitoba, Y. Unno, N. Enrique Yalta Soplin, J. Heymann, M. Wiesner, N. Chen, A. Renduchintala, and T. Ochiai, "ESPnet: End-to-end speech processing toolkit," in *INTERSPEECH*, pp. 2207–2211, 2018.

[29] B. Desplanques, J. Thienpondt, and K. Demuynck, "ECAPA-TDNN: emphasized channel attention, propagation and aggregation in TDNN based speaker verification," in *INTERSPEECH*, pp. 3830–3834, 2020.

[30] A. Nagrani, J. S. Chung, and A. Zisserman, "Voxceleb: a large-scale speaker identification dataset," in *INTERSPEECH*, pp. 2616–2620, 2017.

[31] J. S. Chung, A. Nagrani, and A. Zisserman, "Voxceleb2: Deep speaker recognition," in *INTERSPEECH*, pp. 1086–1090, 2018.

[32] M. Ravanelli, T. Parcollet, P. Plantinga, A. Rouhe, S. Cornell, L. Lugosch, C. Subakan, N. Dawalatabad, A. Heba, J. Zhong, J.-C. Chou, S.-L. Yeh, S.-W. Fu, C.-F. Liao, E. Rastorgueva, F. Grondin, W. Aris, H. Na, Y. Gao, R. D. Mori, and Y. Bengio, "SpeechBrain: A general-purpose speech toolkit," *arXiv preprint arXiv:2106.04624*, 2021.

[33] T. Saeki, D. Xin, W. Nakata, T. Koriyama, S. Takamichi, and H. Saruwatari, "UTMOS: UTokyo-SaruLab System for VoiceMOS Challenge 2022," in *INTERSPEECH*, pp. 4521–4525, 2022.

[34] W.-C. Huang, E. Cooper, Y. Tsao, H.-M. Wang, T. Toda, and J. Yamagishi, "The VoiceMOS Challenge 2022." in *INTERSPEECH*, pp. 4536–4540, 2022.