Interspeech 2024
1-5 September 2024, Kos, Greece

Knowledge Distillation from Self-Supervised Representation Learning Model with Discrete Units for Any-to-Any Streaming Voice Conversion

Hiroki Kanagawa, Yusuke Ijima

NTT Corporation, Japan
hiroki.kanagawa@ntt.com

Abstract

SSL models like HuBERT and WavLM serve as effective content encoders for non-parallel voice conversion (VC), but their large size and design for offline operation make streaming use a challenge. Thus, we derive novel lightweight streaming VC using knowledge distillation (KD) from the SSL model. A promising SSL model and its vector quantizer are used as the teacher content encoder. The student content encoder predicts discrete content from the teacher, ensuring consistency within the KD framework. To stabilize the converted speech's prosody, a prosody predictor using content and speaker information is employed. A HiFi-GAN-like decoder generates waveforms from speaker, content, and prosody inputs. Our student VC leverages the SSL model's robust content encoding without relying on it for inferencing, enabling streaming operation. Evaluations on any-to-any VC tasks show our approach achieved naturalness comparable to modern offline VCs and the teacher with SSL model while being streamable.
Index Terms: streaming voice conversion, knowledge distillation, self-supervised representation learning model

1. Introduction

Voice conversion (VC) aims to change a speaker's voice identity while preserving the speech content. VC training methods are broadly categorized into parallel and non-parallel approaches. As for the VC operation in inferencing, there are two modes: offline and streaming. For practical applications that require real-time performance, the streaming mode is preferred.

Parallel VC can be modeled relatively easily because the speech content is the same for both source and target speakers. However, multi-speaker VC (e.g., any-to-any, many-to-many) requires substantial parallel data across all speakers, making data collection expensive. Non-parallel VC enables learning from large-scale data where the speech content differs between source and target speakers. Most models aim to extract content information by removing speaker identity from the source speech during training, then adding the target speaker's characteristics so as to minimize reconstruction error. Early non-parallel VC approaches like StarGAN-VC [1] and AutoVC [2] use generative adversarial networks (GANs) [3] and variational autoencoders (VAEs) [4] respectively. However, these models lack linguistic constraints, making it difficult to separate content and speaker information.

To address this, some methods explicitly extract content by applying automatic speech recognition (ASR) to the source speaker [5–10]. However, their performance heavily depends on ASR accuracy, and ASR models require text and speech data for training. Other approaches like VITS [11] and its variants (e.g YourTTS [12]) enabled text-to-speech synthesis and voice conversion but also require text data for training, similar to ASR-based methods. VITS allows mutual conversion between its encoder (considering text and speaker) and posterior encoder outputs by using normalizing flows [13].

Recently, self-supervised learning (SSL) models have been used to build robust voice conversion without transcribed text. SoftVC [14] uses HuBERT [15] as a content encoder, SSL outputs are converted into discrete units to partially remove speaker identity. Its performance can be improved by projecting the SSL output into the probability distribution of these units. FreeVC [16] achieves better performance without transcribed text by replacing VITS's content encoder with an SSL model. However, SSL-based content encoders are computationally intensive non-autoregressive (non-AR) models designed for offline operation. While QuickVC [17] improved operational speed with an inverse Fourier transform vocoder [18], attaining streaming operation with SSL models remains challenging.

This work proposes an approach that uses knowledge distillation (KD) from SSL models for streaming non-parallel VC; it leverages SSL in streaming operation. In contrast to other streaming VC models [19–23] that rely on non-end-to-end methods or avoid SSL due to its complexity, the proposed method uses a lightweight streaming content encoder learned by distilling knowledge from an SSL-based teacher VC model. Inspired by vector quantization (VQ) approaches [14, 24], we use discrete units without speaker identity as teacher content. The decoder generating speech waveforms from content and speaker information is optimized end-to-end with the content encoder to prevent performance from being degraded by content prediction errors. Experiments show the proposed model matches the performance of modern offline VC models like VITS and FreeVC while supporting streaming operation[1].

2. Proposed streaming any-to-any VC via knowledge distillation from SSL model

2.1. Model architecture and its training

Our proposed framework aims to leverage the robust content extraction capability of SSL models in streaming VC. Figure 1(a) overviews our proposed method's training diagram. This is based on KD, where a compact student VC model learns to mimic the content representations produced by a larger, SSL teacher model. The overall structure of the proposed method consists of the following key components:

Teacher content encoder (Fig. 1(b)): This is a pre-trained SSL model coupled with a VQ. It takes raw speech waveforms as input and produces discrete content representations.

[1] Sample audios are available here:
https://hkanagawa.github.io/interspeech2024npvc/

10.21437/Interspeech.2024-924

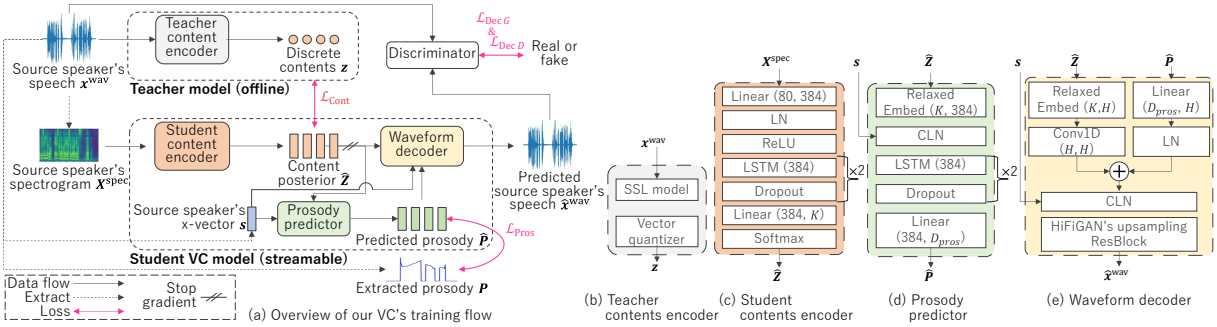

Figure 1: *The training diagram for our streaming VC based on KD from a teacher content encoder using the SSL model. LN and CLN stand for layer normalization and conditional LN modules, respectively. Each weighted module is tagged with input and unit size. We used 3 and 512 as the prosodic features' dimension D_{Pros} and HiFi-GAN's initial channel H, respectively.*

Student VC Model: This is the main component being trained. This model has: I) Student content encoder: Learns to extract content posterior $\hat{\boldsymbol{Z}}$ from mel-spectrograms $\boldsymbol{X}^{\text{spec}}$ instead of waveforms $\boldsymbol{x}^{\text{wav}}$ to reduce input length (Fig. 1(c)). II) Prosody predictor: Predicts prosodic features based on content posterior $\hat{\boldsymbol{Z}}$ and speaker information $\boldsymbol{s}$ (Fig. 1(d)). III) Waveform decoder: Generates speech waveforms $\hat{\boldsymbol{x}}^{\text{wav}}$ from $\hat{\boldsymbol{Z}}$, $\boldsymbol{s}$, and $\hat{\boldsymbol{P}}$ (Fig. 1(e)). The training process works as follows:

1. The student content encoder is trained to match the discrete contents, $\boldsymbol{z}$, produced by the teacher content encoder, using $\mathcal{L}_{\text{Cont}} = \text{CrossEntropy}\left(\boldsymbol{z}, \hat{\boldsymbol{Z}}\right)$.
2. The prosody predictor is trained to predict accurate prosodic features $\hat{\boldsymbol{P}}$ from the content and speaker embeddings $\boldsymbol{s}$, using $\mathcal{L}_{\text{Pros}} = \text{L1Loss}\left(\boldsymbol{P}, \hat{\boldsymbol{P}}\right)$.
3. The waveform decoder is trained to generate high-fidelity waveforms from the content and prosody representations, using discriminator and generator losses from HiFi-GAN, $\mathcal{L}_{\text{DecD}}$ and $\mathcal{L}_{\text{DecG}}$, respectively.

To train these modules, the final loss is actually given by:

$$\mathcal{L}_{\text{VC}} = \mathcal{L}_{\text{Cont}} + \mathcal{L}_{\text{Pros}} + \mathcal{L}_{\text{DecG}}. \quad (1)$$

Thus, the student VC model and the discriminator are alternately updated by $\mathcal{L}_{\text{VC}}$ and $\mathcal{L}_{\text{DecD}}$, respectively.

The key advantage of this framework is that it distills the powerful content extraction capabilities of large SSL models into a compact, streamable VC model. The student model can directly obtain content representations without running the computationally expensive SSL teacher in inferencing. To allow streaming operation, the student model components use unidirectional LSTMs and conditional layer normalization [25] for speaker conditioning. A pseudo-stream vocoding approach also is employed as described in the next section. Overall, our approach is expected to work well because it leverages the robust content representations by SSL models, while producing a compact and efficient VC system tailored for streaming applications.

2.2. Pseudo-stream vocoding with non-AR waveform decoder

We use a [26]-like HiFi-GAN variant as the waveform decoder. In preliminary experiments, we tried to fully causalize the HiFi-GAN as described in [22], but could not achieve comparable quality to the original HiFi-GAN. Therefore, multiple frames separated into chunks are given to the non-causal HiFi-GAN to achieve pseudo-stream vocoding. Figure 2 illustrates our pseudo-stream vocoding flow. We vocode waveforms within N frame chunks as a single chunk, and overlap the contents of

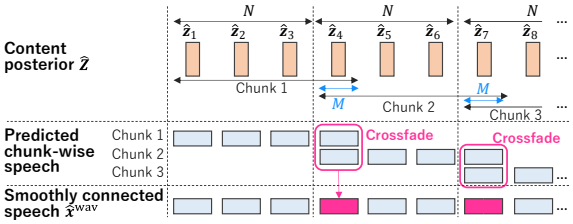

Figure 2: *Chunk-wise pseudo-stream vocoding with crossfade. M and N are the number of overlap frames and chunk size, respectively. Overlapped segments outlined in pink are smoothly connected by crossfade. For clarity, prosody is omitted.*

the M frames to avoid discontinuities that could cause click artifacts. These overlapped segments are connected smoothly by crossfading.

3. Experiments

3.1. Setup

We used two types of Japanese multi-speaker speech databases. The first database was for model training and evaluation on seen speakers, containing 5,284 speakers and a substantial number of utterances (total of 588.3 hours). From this, fifty utterances each from four male and female speakers, eight speakers in total, were extracted for evaluation on seen speakers, and the remaining utterances were used for training along with other speakers. The second database was for evaluation on unseen speakers, consisting of four male and female speakers, eight speakers in total, with each speaker having fifty utterances (about 2.8 minutes) similar to the seen speakers. From both databases, two male and female speakers were assigned as source and target speakers for same-gender and cross-gender combinations, respectively. The sampling rate was 24 kHz. As the input acoustic features for the proposed method, we employed eighty-dimensional logarithmic mel-spectrograms with a 20 ms frame shift to match the the SSL model's output. The three-dimensional prosody feature consists of z-scored log-F0 and energy and a binary voice-unvoiced flag. For the speaker conditioning features, all methods used 256-dimensional x-vectors. The FastResNet34-based x-vector extractor was pre-trained on speech from over 8,000 Japanese speakers, including the training data [27].

Table 1 lists the compared methods. VITS implementation is employed for the ideal VC condition when both speech and transcribed text are available [28]. We fed 380 kinds of symbols, including phoneme and prosodic information, to VITS. FREEVC implementation substitutes VITS's transcribed text

Table 1: *Comparison VC models and their features.*

Method	Training data	Use SSL for training	Run SSL for inference	Streamable
VITS [11]	Text/speech pairs	No	No	No
FREEVC [16]	Speech	Yes	Yes	No
TEACHERCONTENT	Speech	Yes	Yes	No
KDOFFLINE (ours)	Speech	Yes	No	No
KDSTREAM (ours)	Speech	Yes	No	Yes

with SSL model's output [29]. In original FREEVC, although the output audio's sampling rate was 16 kHz, we set it to 24 kHz to match the other methods. As is mentioned in Section 3.2.2, preliminary investigatations of the word error rate (WER) found that using HuBERT-base (Japanese) [30] for the content encoder yielded better results than WavLM-large (English) [31, 32] in our Japanese VC tasks, and so it was chosen. TEACHERCONTENT obtains discrete contents by feeding audio into an SSL model and discretizing its output. In other words, this method's content encoder acts as the teacher in our proposed method. We also trained not only the content encoder but also 1) the prosody encoder based on bi-directional LSTM using the extracted content and x-vector, and 2) the waveform decoder with the extracted content, prosody features, and x-vector. Our proposed method has two variants, KDOFFLINE and KDSTREAM, using bi- and uni-directional LSTM for the content encoder and prosody predictor, respectively. For the proposed method, the same waveform decoder architecture of TEACHERCONTENT was used. As for KDSTREAM's waveform decoder, the parameters for pseudo-stream vocoding described in Section. 2.2 were set to $M = 1$ and $N = 9$. All these VC models were optimized in 1,500 k steps with a batch size of 32 by Adam [33] with a fixed learning rate of 2×10^{-4}.

3.2. Objective evaluations

3.2.1. Parametric study for the number of discrete classes, K

While prior work SoftVC mentions that speaker characteristics can be removed by discretizing SSL features, no discussion of the connection between the number of classes, K, and VC performance was mentioned. Thus, we first check the performance when K is varied in TEACHERCONTENT using the objective measures of WER, mel-cepstrum distortion (MCD), and F0 root mean square error (RMSE); these cover the three aspects of reproduction, linguistic content, spectral, and prosodic, respectively. WER was calculated for the recognition hypotheses obtained from Whisper (large-v2) [34]. MCD was obtained by converting the mel-spectrogram from VC into a 40-dimensional mel-cepstrum and then aligning the sequence length with that of the target speaker's natural speech by DTW. For F0 RMSE calculation, we first extracted the F0 from the converted speech using the RAPT [35]. Then, after aligning it with the target speaker's extracted F0 using the DTW paths from the MCD calculation, the RMSE across both voiced segments was calculated. Figure 3 shows the objective evaluation results with "Seen-to-Seen" speaker pair setting. The horizontal axis is the number of discrete class K, and the vertical axes in Fig.3 (a)-(c) of the figure plot WER, MCD, and F0 RMSE, respectively. The results demonstrate that the WER improved as the value of K increased. This is because higher values of K enable richer linguistic expressiveness within the content representations. While no critical degradation was observed in MCD, F0 RMSE degraded as K increased. This degradation occurs because by increasing K, not only the linguistic expression but also speaker information becomes encoded within the content, making it more challenging to accurately reproduce the target

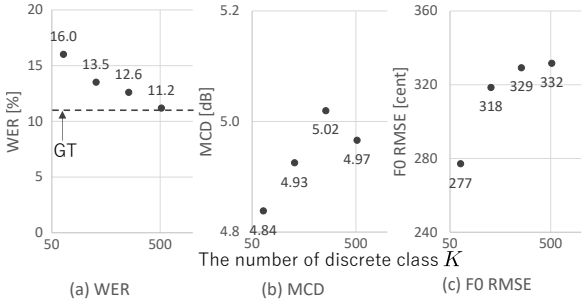

Figure 3: *The objective evaluation results of* TEACHERCONTENT *in the "Seen-to-Seen" speaker pair setting. The number of discrete classes, K, is 64, 128, 256, and 512, and the WER, MCD and F0 RMSE are shown in (a)–(c), respectively. The WER of the target speaker's original utterances (*GT*) is also listed, serving as the used ASR's upper limit.*

Table 2: *Averaged WER, MCD, and F0 RMSE from evaluation data for all speaker pair settings. The scores written in bold signify the column-wise best value except* GT.

Method	WER [%]	MCD [dB]	F0 RMSE [cent]
(Reference) GT	10.4	-	-
VITS	20.8	5.40	**219**
FREEVC	13.8	6.30	407
TEACHERCONTENT	**11.6**	**5.08**	317
KDOFFLINE (ours)	13.1	5.14	309
KDSTREAM (ours)	13.0	5.13	323

speaker's prosody. However, since we take the preservation of the linguistic content of the converted speech to have the highest priority for voice conversion, our proposed methods were implemented with $K = 512$, which exhibited the best WER comparable to GT's, and compared it with those of other methods in the subsequent evaluation.

3.2.2. Comparison of Objective Measures

We evaluate the proposed method using objective metrics based on the number of discrete classes, K, from the previous section. Table 2 shows the objective scores across evaluation data for all speaker pair settings. Detailed results for each speaker pair setting will be discussed later in the subjective evaluation (next section) due to space constraints. To establish an upper bound on speech recognition performance, we evaluate the WER against ground truth transcriptions. VITS had the worst WER due to its lack of an SSL model for voice conversion, resulting in weak content encoding. However, VITS achieved the best F0 RMSE score among all methods because it uses transcribed text during training, which prevents source speaker characteristics from leaking into the extracted content during inference. FREEVC leverages robust SSL-based content representations, giving it a good WER. However, it attained the worst MCD and F0 RMSE scores, suggesting insufficient removal of source speaker traits from the content. TEACHERCONTENT achieves the best WER and MCD performance by effectively removing speaker characteristics via the combination of an SSL model and VQ; this agrees with prior work. Our proposed KDOFFLINE method slightly underperforms TEACHERCONTENT in WER but achieves comparable performance without running the SSL model in inferencing. The streaming version, KDSTREAM, maintains similar WER and MCD to KDOFFLINE, with a slight drop in F0 RMSE. These results demonstrate that our proposed method can effectively mimic the teacher voice conversion model's content representations without relying on

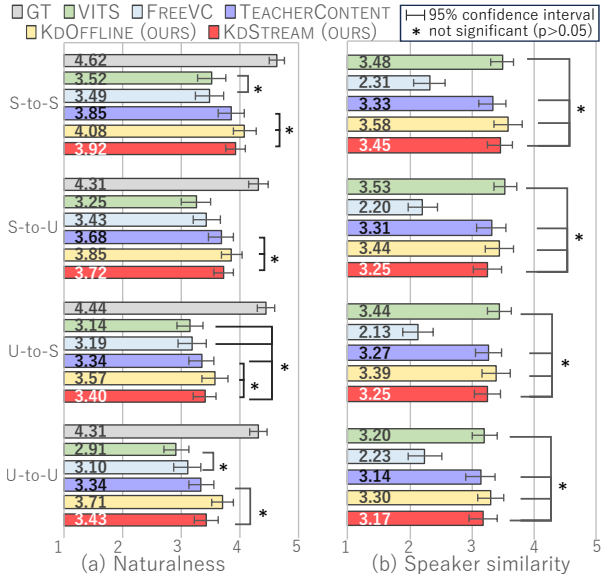

Figure 4: *Subjective evaluation results. For example, "S-to-U" denotes the speaker pair setting conversion from "seen" (S) source speaker to "unseen" (U) target.*

the SSL model at inference time.

3.3. Subjective evaluations

We subjectively evaluated naturalness and speaker similarity for each method using 5-scale mean opinion scores (MOS) and degradation MOS (DMOS). Sixteen Japanese listeners participated in this test. For each method, each listener first listened to 16 randomly selected speech samples and rated their naturalness on a scale from 1: very unnatural to 5: very natural. Similarly, they rated the speaker similarity of each method's voice sample to the target speaker's ground truth audio, which was played as a reference, on a scale from 1: very dissimilar to 5: very similar. Unlike the objective evaluation in Section 3.2.2, the scores were aggregated for each combination of the source and target speakers being "seen" (S) or "unseen" (U) (e.g., "S-to-U" is the conversion setting from a seen speaker to an unseen speaker).

Figure 4 shows the subjective evaluation results. Except for the "S-to-U" case, VITS showed approximately the same naturalness as FreeVC. Since VITS was the only SSL-free method among the compared VCs, it sometimes yielded ambiguous utterances or mispronunciations. In particular, the "U-to-U" score was significantly lower, indicating that performance degrades when both the source and target speakers are unseen. While FreeVC learned without text, it was comparable to or slightly better than VITS in naturalness, confirming the robust content extraction capability of the SSL models. However, FreeVC performed the worst among all methods in regard to speaker similarity. FreeVC got the worst F0 RMSE score among all methods in Section 3.2.2, suggesting poor removal of speaker characteristics from the content. On the other hand, TeacherContent, which uses the same SSL model as FreeVC, was superior to FreeVC's naturalness in many speaker settings. Its speaker similarity was also comparable to VITS among all methods, confirming that the content discretization can properly remove speaker characteristics. Among our proposed methods, KdOffline achieved the best naturalness among all VC methods while matching the speaker similarity of VITS and TeacherContent. Unlike TeacherContent, our proposed method used the content's posterior probability as soft labels. All modules, including the content encoder (with errors), prosody predictor and the waveform decoder, were trained in an end-to-end manner, which likely contributed to the improvement. Furthermore, KdStream achieved similar speaker similarity to VC methods other than FreeVC, while achieving naturalness only slightly degraded from KdOffline and comparable to TeacherContent. These results demonstrated that the proposed method could robustly capture content, the strength of SSL models, while enabling streaming operation as the SSL model is not used for inferencing.

Table 3: *Average RTFs for content, prosody, and waveform generation, and their totals obtained from all evaluation data. The scores written in bold signify the column-wise best. Note that among these, only our* KdStream *is capable of streaming operation.*

Method	Contents	Prosody	Waveform	Total
VITS	0.041	-	0.440	0.481
FreeVC	0.194	-	0.444	0.638
TeacherContents	0.196	**0.004**	0.398	0.598
KdOffline (ours)	**0.004**	**0.004**	0.398	0.407
KdStream (ours)	0.012	0.013	**0.376**	**0.402**

3.4. Speed comparison

We compared the speed of our method with those of other methods. All speed tests were done using an Intel Core i9-9920X CPU 3.5 GHz. The speed metric used was the real-time factor (RTF) for single-threaded operation. The RTF is calculated as $T_{\text{inference}}/T_{\text{wav}}$, where $T_{\text{inference}}$ is the time taken for inference and T_{wav} is the converted speech length. Table 3 shows the total RTF and its breakdown for each method. Focusing on the RTF for content processing first, FreeVC and TeacherContent, which run the SSL model, had higher RTFs compared to VITS. In contrast, our KdOffline and KdStream were significantly faster than the others. KdOffline was faster than KdStream for content processing because the LSTM computations were done in batches instead of frame-by-frame. The prosody predictor was similar in scale to the content encoder, so their RTFs were about the same. For the total RTF including waveform generation, our KdStream was the fastest. The main reason for the fast vocoding of KdStream was that efficient CNN computations could be used because the chunk size remained constant in each forward pass. KdStream's chunk size is $M + N$, from Section 2.2. According to [23], its latency given by $(M + N) \times \text{hop\_size} \times (1 + \text{RTF})$ was 0.28 seconds, with a hop size of 480. While the overall RTF for speech waveform generation is still high, speeds can be improved further by using a lightweight vocoder as in [18, 36]. The latency can also be further reduced by choosing an optimal N or by using a decoder based on streaming-friendly neural audio codecs suitable for VC [37].

4. Conclusion

In this study, we proposed a new approach to realize streaming non-parallel VC by leveraging knowledge distillation from SSL models. Our proposed VC model distills the powerful content extraction capabilities of large-scale SSL models into a compact, streamable VC model, without the need to run the SSL model in inferencing. Experimental demonstrated that our proposed VC model matches the level of naturalness achieved by modern offline VC models while supporting streaming operations.

5. References

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

[2] K. Qian, Y. Zhang, S. Chang, X. Yang, and M. Hasegawa-Johnson, "AutoVC: Zero-shot voice style transfer with only autoencoder loss," *Proc. ICML*, vol. 97, pp. 5210–5219, 2019.

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

[4] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *Proc. ICLR*, 2014.

[5] L. Sun, K. Li, H. Wang, S. Kang, and H. Meng, "Phonetic posteriorgrams for many-to-one voice conversion without parallel data training," in *Proc. ICME*, 2016, pp. 1–6.

[6] H. Miyoshi, Y. Saito, S. Takamichi, and H. Saruwatari, "Voice conversion using sequence-to-sequence learning of context posterior probabilities," *Proc. Interspeech*, pp. 1268–1272, 2017.

[7] S. Liu, Y. Cao, D. Wang, X. Wu, X. Liu, and H. Meng, "Any-to-many voice conversion with location-relative sequence-to-sequence modeling," *IEEE Trans. on Audio, Speech, and Language Processing*, vol. 29, pp. 1717–1728, 2021.

[8] W.-C. Huang, T. Hayashi, Y.-C. Wu, H. Kameoka, and T. Toda, "Pretraining techniques for sequence-to-sequence voice conversion," *IEEE Trans. Audio, Speech, and Language Processing*, vol. 29, pp. 745–755, 2021.

[9] S. Ding, G. Zhao, and R. Gutierrez-Osuna, "Improving the Speaker Identity of Non-Parallel Many-to-Many Voice Conversion with Adversarial Speaker Recognition," *Proc. Interspeech*, pp. 776–780, 2020.

[10] X. Zhao, F. Liu, C. Song, Z. Wu, S. Kang, D. Tuo, and H. Meng, "Disentangling content and fine-grained prosody information via hybrid asr bottleneck features for voice conversion," *Proc. ICASSP*, pp. 7022–7026, 2022.

[11] J. Kim, J. Kong, and J. Son, "Conditional variational autoencoder with adversarial learning for end-to-end text-to-speech," *Proc. ICML*, vol. 139, pp. 5530–5540, 2021.

[12] 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 voice conversion for everyone," *Proc. ICML*, pp. 2709–2720, 2022.

[13] D. Rezende and S. Mohamed, "Variational inference with normalizing flows," *Proc. ICML*, pp. 1530–1538, 2015.

[14] 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," *Proc. ICASSP*, pp. 6562–6566, 2022.

[15] 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 Trans. Speech and Audio Processing*, vol. 29, pp. 3451–3460, 2021.

[16] J. Li, W. Tu, and L. Xiao, "FreeVC: Towards high-quality text-free one-shot voice conversion," *Proc. ICASSP*, pp. 1–5, 2023.

[17] H. Guo, C. Liu, C. T. Ishi, and H. Ishiguro, "QUICKVC: A lightweight VITS-based any-to-many voice conversion model using ISTFT for faster conversion," *Proc. ASRU*, pp. 1–7, 2023.

[18] M. Kawamura, Y. Shirahata, R. Yamamoto, and K. Tachibana, "Lightweight and high-fidelity end-to-end text-to-speech with multi-band generation and inverse short-time fourier transform," *Proc. ICASSP*, pp. 1–5, 2023.

[19] R. Arakawa, S. Takamichi, and H. Saruwatari, "Implementation of dnn-based real-time voice conversion and its improvements by audio data augmentation and mask-shaped device," *Proc. SSW*, pp. 93–98, 2019.

[20] H. Kameoka, K. Tanaka, and T. Kaneko, "FastS2S-VC: Streaming non-autoregressive sequence-to-sequence voice conversion," *arXiv preprint arXiv:2104.06900*, 2022.

[21] T. Hayashi, K. Kobayashi, and T. Toda, "An investigation of streaming non-autoregressive sequence-to-sequence voice conversion," *Proc. ICASSP*, pp. 6802–6806, 2022.

[22] Z. Chen, H. Miao, and P. Zhang, "Streaming non-autoregressive model for any-to-many voice conversion," *arXiv preprint arXiv:2206.07288*, 2022.

[23] Z. Ning, Y. Jiang, P. Zhu, J. Yao, S. Wang, L. Xie, and M. Bi, "DualVC: Dual-mode voice conversion using intra-model knowledge distillation and hybrid predictive coding," *Proc. Interspeech*, pp. 2063–2067, 2023.

[24] D. Wang, L. Deng, Y. T. Yeung, X. Chen, X. Liu, and H. Meng, "VQMIVC: Vector Quantization and Mutual Information-Based Unsupervised Speech Representation Disentanglement for One-Shot Voice Conversion," *Proc. Interspeech*, pp. 1344–1348, 2021.

[25] M. Chen, X. Tan, B. Li, Y. Liu, T. Qin, S. Zhao, and T.-Y. Liu, "AdaSpeech: Adaptive text to speech for custom voice," *Proc. ICLR*, 2021.

[26] A. Polyak, Y. Adi, J. Copet, E. Kharitonov, K. Lakhotia, W.-N. Hsu, A. Mohamed, and E. Dupoux, "Speech resynthesis from discrete disentangled self-supervised representations," *Proc. Interspeech*, pp. 3615–3619, 2021.

[27] J. S. Chung, J. Huh, S. Mun, M. Lee, H.-S. Heo, S. Choe, C. Ham, S. Jung, B.-J. Lee, and I. Han, "In defence of metric learning for speaker recognition," *Proc. Interspeech*, pp. 2977–2981, 2020.

[28] [Online] https://github.com/jaywalnut310/vits

[29] [Online] https://github.com/OlaWod/FreeVC

[30] [Online] https://huggingface.co/rinna/japanese-hubert-base

[31] 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.

[32] [Online] https://huggingface.co/microsoft/wavlm-large

[33] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *Proc. ICLR*, 2015.

[34] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. Mcleavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," *Proc. ICML*, vol. 202, pp. 28 492–28 518, 2023.

[35] D. Talkin and W. B. Kleijn, "A robust algorithm for pitch tracking (RAPT)," *Speech coding and synthesis*, vol. 495, p. 518, 1995.

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

[37] Y. Yang, Y. Kartynnik, Y. Li, J. Tang, X. Li, G. Sung, and M. Grundmann, "StreamVC: Real-time low-latency voice conversion," *arXiv preprint arXiv:2401.03078*, 2024.