Neural Processing Letters (2024) 56:166
https://doi.org/10.1007/s11063-024-11613-0

WaveVC: Speech and Fundamental Frequency Consistent Raw Audio Voice Conversion

Kyungdeuk Ko[1] · Donghyeon Kim[1] · Kyungseok Oh[1] · Hanseok Ko[1]

Accepted: 6 April 2024

Abstract
Voice conversion (VC) is a task for changing the speech of a source speaker to the target voice while preserving linguistic information of the source speech. The existing VC methods typically use mel-spectrogram as both input and output, so a separate vocoder is required to transform mel-spectrogram into waveform. Therefore, the VC performance varies depending on the vocoder performance, and noisy speech can be generated due to problems such as train-test mismatch. In this paper, we propose a speech and fundamental frequency consistent raw audio voice conversion method called WaveVC. Unlike other methods, WaveVC does not require a separate vocoder and can perform VC directly on raw audio waveform using 1D convolution. This eliminates the issue of performance degradation caused by the train-test mismatch of the vocoder. In the training phase, WaveVC employs speech loss and F0 loss to preserve the content of the source speech and generate F0 consistent speech using the pre-trained networks. WaveVC is capable of converting voices while maintaining consistency in speech and fundamental frequency. In the test phase, the F0 feature of the source speech is concatenated with a content embedding vector to ensure the converted speech follows the fundamental frequency flow of the source speech. WaveVC achieves higher performances than baseline methods in both many-to-many VC and any-to-any VC. The converted samples are available online.

Keywords Voice conversion · Adversarial training · Deep learning

1 Introduction

Style transfer is applied in various fields, including vision tasks [1]. Especially in the field of speech signal processing [2, 3], voice conversion (VC) is a task for changing the speech of

✉ Hanseok Ko
hsko@korea.ac.kr

Kyungdeuk Ko
kdko@korea.ac.kr

Donghyeon Kim
kis6470@korea.ac.kr

Kyungseok Oh
lshh0000@korea.ac.kr

[1] School of Electrical Engineering, Korea University, 02841 Seoul, South Korea

Published online: 08 May 2024

a source speaker to the target speaker's voice while preserving linguistic information of the source speech The application of VC has the potential for utilization in various fields such as movie dubbing, singing conversion [4], and speaking aids [5]. Typically, conventional VC methods require parallel data, which are recordings of different speakers saying the same sentence. However, obtaining such parallel data is an obvious limitation as it is very difficult in practice. For this reason, various methods using non-parallel data for VC have recently been explored.

Autoencoder-based VC methods [6–8] utilize zero-shot learning to enable the use of unparallel data for training. These methods typically consist of a content encoder, a speaker encoder, and a decoder. While these methods are relatively easy to train, they must be carefully designed to disentangle the content and the style well with a bottleneck structure. To compensate for these shortcomings, vector quantization (VQ) is applied to VC. In the VQ-based VC methods [9–11], the discrete content embedding vector is generated by the VQ of the continuous content embedding vector. Then, the speaker embedding vector is defined by the difference between the continuous content embedding vector and the discrete content embedding vector. However, VQ causes a lot of information loss, such as time relationships and fundamental frequency, which leads to performance degradation. Generative adversarial network (GAN) [12, 13] is applied to VC for the quality improvement of the converted speech. For example, StarGAN [14]-based VC methods [15–17] generate high-quality speech using adversarial training and perceptual loss.

However, these StarGAN-based methods have crucial limitations in that they can't respond to unseen target speakers. Also, in the existing VC methods, including autoencoder-based VC and GAN-based methods, vocoders such as MelGAN [18], Parallel WaveGAN [19], and HiFi-GAN [20] are required to transform the converted mel-spectrogram into the raw audio waveform. Using vocoders can cause problems such as noisy speech generation for reasons such as train-test mismatch [21]. When training voices for a new domain using the VC method, a vocoder must be additionally trained. When using a pre-trained vocoder, the input mel-spectrogram hyperparameters of the VC method depend on the pre-trained vocoder. In addition, the existing VC methods focus on disentangling the content and the speech information and generating realistic sounds. Therefore, they don't consider detailed source speech information, such as fundamental frequency and pronunciation. Meanwhile, since only the identity of the speaker is changed to the target speaker, it is not only crucial to keep the fundamental frequency of the source speech consistent, but it is also important to pronounce the speech with high accuracy from an application perspective.

In this paper, we propose a speech and fundamental frequency consistent raw audio waveform VC method called WaveVC. Because WaveVC is composed of 1D-convolutional layers and performs VC directly on the raw audio waveform, it is not affected by vocoder performance. In the training phase, WaveVC employs speech loss and F0 loss to preserve the content of the source speech and generate F0 consistent speech using the pre-trained networks. In the test phase, the F0 feature of the source speech is concatenated with a content embedding vector to ensure the converted speech follows the fundamental frequency flow of the source speech. Our main contributions are summarized as follows: (1) WaveVC performs VC directly from the raw audio waveform. An additional vocoder is not required to convert the mel-spectrogram to the raw audio waveform. (2) In the training phase, WaveVC employs two additional losses: one is speech consistency loss, and the other is F0 consistency loss. The consistency losses preserve content information and guarantee fundamental frequency consistency. (3) WaveVC shows higher objective and subjective performance than other VC

methods in many-to-many and any-to-any VC. The converted samples are available on the web demo page.[1]

2 Related Works

2.1 Speech Synthesis

Speech synthesis with a desired target speaker has been studied. WaveNet [22] uses the linguistic features as the input to generate speech. WaveNet employs dilated casual convolution to cover long-rage temporal dependencies. WaveNet can generate various characteristic voices using global conditioning and local conditioning. DeepVoice1 [23] follows the three components of statistical parametric synthesis, and Tacotron [24] proposes an attention-based seq-to-seq model for end-to-end speech synthesis. DeepVoice2 [25] trains speaker embedding and applies it to not only DeepVoice1 but also Tacotron1. Also, DeepVoice1 and DeepVoice2 employ WaveNet as the vocoder and perform better than the Griffin-Lim algorithm. VAE-Tacotron2 [26] employs variational autoencoder [27] for learning latent representation for style control. However, these methods require the vocoder, and as seen from DeepVoice2, they are greatly influenced by the vocoder. ClariNet [28], FastSpeech2s [29], and EATS [30] propose fully end-to-end speech synthesis without the need for the vocoder. The existing speech synthesis methods have limitations in that text information is entered as input, and the desired style cannot be perfectly generated.

2.2 Voice Conversion

Unlike speech synthesis, VC resynthesizes speech using only the source and target speech. The purpose of VC is to convert the speech to the target speaker's voice while preserving the linguistic information. Most VC methods are composed of a content encoder, a speaker encoder, and a decoder to accomplish this purpose. Zero-shot learning-based VC methods are trained to reconstruct the input data. The content encoder erases the style of the source speaker while keeping linguistic information in the utterance. In contrast, the speaker encoder extracts only the style of the target speaker regardless of the utterance. AutoVC [6] applied zero-shot learning to VC for the first time and can respond to unseen speakers not used for training. AdaIN-VC [7] does not simply concatenate the style of the target speaker extracted from the speaker encoder but reflects the style through adaptive instance normalization [31, 32]. AutoVC-F0 [8] uses the F0 information of the source speaker to generate a natural-sounding F0. Again-VC [33] uses only one encoder without the separate speaker encoder, unlike other autoencoder-based methods. Meanwhile, these zero-shot learning-based methods have limitations in that they must be carefully designed to disentangle the content and the style well with a bottleneck structure.

Recently, GAN-based methods such as StarGANv2-VC [17] show high-quality VC performance using adversarial training and perceptual loss. However, StarGANv2-VC has a crucial limitation: it cannot respond to unseen target speakers. Therefore, many efforts are being made on GAN-based any-to-any VC [34–36] to respond to unseen source speakers and unseen target speakers. Since the aforementioned autoencoder-based methods and GAN-based methods both output the acoustic feature like mel-spectrogram, vocoders such as MelGAN [18], Parallel WaveGAN [19], and HiFi-GAN [20] are needed to convert the

[1] https://kyungdeuk.github.io/wavevc-demo/.

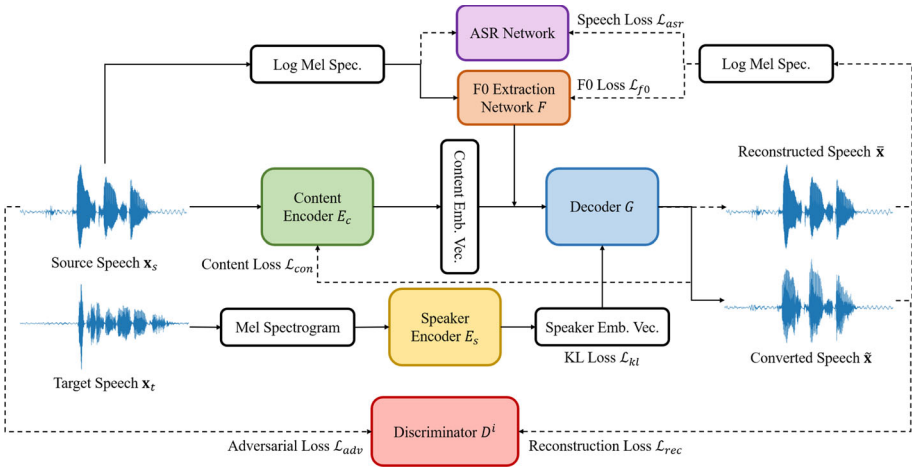

Fig. 1 The overall architecture of WaveVC. The solid lines are the paths used in training and inference, and the dotted lines are used only in training

mel-spectrogram into the raw wave. Using vocoders can cause problems such as noisy speech generation for reasons such as train-test mismatch [21]. As a result, the quality of the generated speech depends on the vocoder. In the VC task, NVC-Net [37] solves the problem of using the vocoder by directly generating raw audio waveform. However, NVC-Net doesn't guarantee that high-quality speech is generated while maintaining the source speech's fundamental frequency.

3 Method

3.1 WaveVC

WaveVC mainly consists of a content encoder E_c, a decoder G, a speaker encoder E_s, three discriminators D^i for $i = 1, 2, 3$ that are used for different temporal resolutions, and an F0 extraction network F. The overall architecture of WaveVC is shown in Fig. 1.

Content encoder Since the input of the content encoder E_c is a raw audio waveform, the content encoder E_c consists of one input 1D convolutional layer, four downsampling blocks, and two following 1D convolutional layer, where kernel size is 7 and padding size is 3, with GELU activation [38]. Each downsampling block consists of four residual blocks and a 1D convolutional layer. Each residual block has a 1D dilated convolutional layer with a gated-tanh nonlinear function and residual skip connection. Figure 2a illustrates the residual block of the content encoder. Each downsampling block makes the input of the block four times the lower temporal resolution. Finally, the source waveform $\mathbf{x}$ has a temporal resolution that is 256 times lower by the content encoder E_c, and $L2$ norm is applied to the content embedding vector.

Speaker encoder Unlike the content encoder E_c, the speaker encoder E_s uses mel-spectrogram as the input. The speaker encoder E_s consists of five residual blocks and a global average pooling layer, and a 512-dimensional vector is generated regardless of the input length by removing temporal dimensions. Mean vector $\boldsymbol{\mu}$ and covariance vector $\boldsymbol{\sigma}$

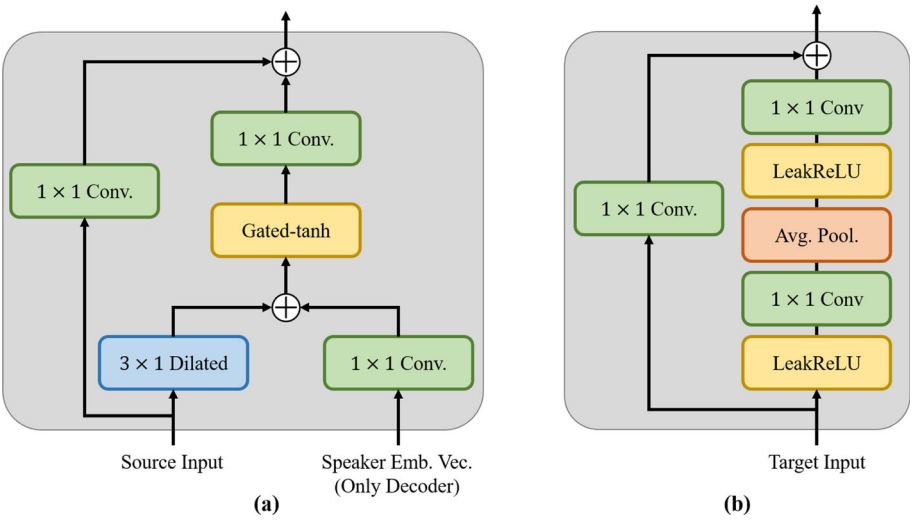

Fig. 2 The detailed architectures. **a** The residual block of the content encoder and the decoder, and **b** the residual block of the speaker encoder

are generated by each fully connected layer. Figure 2 illustrates the residual block of the speaker encoder. The channel size is doubled after each residual block until it reaches 512. Finally, the speaker embedding vector $\mathbf{z}$ is produced by reparameterization trick [27] such as $\mathbf{z} \sim \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \epsilon$, where $\epsilon \sim (\mathbf{0}, \boldsymbol{I})$.

F0 extraction network A pre-trained JDC network [39] composed of the convolutional layers, and bidirectional LSTM is used as the F0 extraction network F to extract the fundamental frequency information. The JDC network is pre-trined jointly with fundamental frequency prediction and voice activity detection. The JDC network uses the mel-spectrogram as the input and outputs the fundamental frequency. Then, the only convolutional layer F_{conv} of the JDC network is used for the F0 information feature extraction. Finally, the F0 information feature $\mathbf{f}$ is defined as $F_{conv}(\mathbf{x})$.

Decoder The decoder G is constructed in the form of an inversion of the content encoder E_c. The decoder G uses the concatenated feature of the content embedding vector ($\mathbf{c}_s = E_c(\mathbf{x}_s)$) and the F0 information feature ($\mathbf{f}_s = F_{conv}(\mathbf{x}_s)$) of the source speech $\mathbf{x}_s$ as the input. The decoder G consists of four upsampling blocks instead of the downsampling blocks. Each upsampling block contains a 1D transposed convolutional layer and four residual blocks. Figure 2a illustrates the residual block of the decoder. Unlike the residual block of the content encoder E_c, the residual block of the decoder G uses the speaker embedding vector as the conditional input. The 1D transposed convolutional layer of the upsampling block makes the input of the upsampling block four times higher in temporal resolution. Then, $\mathbf{z}_s$ and $\mathbf{z}_t$ are used as the speaker embedding vector for the source speech and the target speech.

Discriminator As with MelGAN [18], three discriminators D^i for $i = 1, 2, 3$ use the mel-spectrograms with three different window sizes as the input. Each window size is set to 1024, 512, and 256. The number of speakers in the training data defines the output vector size of the discriminators. Finally, discriminators distinguish whether the input is the corresponding speaker by binary classification.

3.2 Training Objectives

WaveVC aims to accomplish speech and F0 consistent raw audio VC. The losses to achieving this goal are explained. In the following brief description, the reconstructed speech ($\bar{\mathbf{x}} = G(\mathbf{c}_s, \mathbf{z}_s, \mathbf{f}_s)$) and the converted speech ($\tilde{\mathbf{x}} = G(\mathbf{c}_s, \mathbf{z}_t, \mathbf{f}_s)$) are defined, respectively.

Adversarial loss When the speaker id labels of the source speech $\mathbf{x}_s$ and the target speech $\mathbf{x}_t$ are y_s and y_t, respectively, the adversarial loss is defined as

$$\mathcal{L}_{adv} = \mathbb{E}_{\mathbf{x}_s, y_s} \sum_i [\log(D^i(\mathbf{x}_s, y_s))] + \mathbb{E}_{\tilde{\mathbf{x}}, y_t} \sum_i [\log(1 - D^i(\tilde{\mathbf{x}}, y_t)]. \tag{1}$$

Each discriminator D^i is trained through binary classification to distinguish whether it is the corresponding label's speech. Conversely, the content encoder E_c, the speaker encoder E_s, and the decoder G are trained to be indistinguishable from the discriminators D^i.

Speech loss Speech loss is used to ensure that the content of the converted speech is maintained. The speech loss is composed of the differences between the source speech and the converted speech and between the source speech and the reconstructed source speech and is defined as

$$\mathcal{L}_{asr} = \mathbb{E}_{\mathbf{x}_s, \bar{\mathbf{x}}}[\|A(\mathbf{x}_s) - A(\bar{\mathbf{x}})\|_1] + \mathbb{E}_{\mathbf{x}_s, \tilde{\mathbf{x}}}[\|A(\mathbf{x}_s) - A(\tilde{\mathbf{x}})\|_1], \tag{2}$$

where $\|\cdot\|_1$ denotes $l1$ norm. In addition, A is a pre-trained automatic speech recognition (ASR) network for extracting the convolutional speech features from the source speech and the converted speech. In this case, a joint CTC-attention VGG-BLSTM network [40] is employed as the pre-trained ASR network for the speech convolutional feature extraction.

F0 loss F0 loss is used to generate fundamental frequency consistent results. The final output of the F0 extraction network is used as the predicted fundamental frequency. F0 loss is calculated by the differences in the normalized fundamental frequency between the source speech and the converted speech and between the source speech and the reconstructed source speech as follows

$$\mathcal{L}_{f0} = \mathbb{E}_{\mathbf{x}_s, \bar{\mathbf{x}}}[\left\| \hat{F}(\mathbf{x}_s) - \hat{F}(\bar{\mathbf{x}}) \right\|_1] + \mathbb{E}_{\mathbf{x}_s, \tilde{\mathbf{x}}}[\left\| \hat{F}(\mathbf{x}_s) - \hat{F}(\tilde{\mathbf{x}}) \right\|_1], \tag{3}$$

where $\hat{F}(\cdot)$ means the normalized output of the F0 extraction network.

Reconstruction loss Reconstruction loss is composed of two parts to improve perceptual quality. One is feature matching loss [18], and the other is spectral loss [37]. The feature matching loss is calculated by feature maps of the discriminators D^i as follows

$$\mathcal{L}_{fm} = \mathbb{E}_{\mathbf{x}_s, \bar{\mathbf{x}}} \sum_i \sum_j \frac{1}{N_D} \left\| D_j^i(\mathbf{x}_s) - D_j^i(\bar{\mathbf{x}}) \right\|_1, \tag{4}$$

where $D_j^i(\cdot)$ denotes the jth feature map of the ith discriminator, and N_D indicates the number of discriminators. On the other hand, the spectral loss is calculated from mel-spectrograms with different FFT sizes and is defined as

$$\mathcal{L}_{sp} = \mathbb{E}_{\mathbf{x}_s, \bar{\mathbf{x}}} \sum_w \|T(\mathbf{x}_s, w) - T(\bar{\mathbf{x}}, w)\|_2^2, \tag{5}$$

where $T(\cdot, w)$ denotes transformation to log mel-spectrogram with a FFT size of w, and $\|\cdot\|_2$ indicates $l2$ norm. In this case, w is set to 2048, 1024, and 512. Finally, the reconstruction

loss is defined as

$$\mathcal{L}_{rec} = \mathcal{L}_{fm} + \mathcal{L}_{sp}. \tag{6}$$

Content loss Content loss induces that the content embedding vector of the converted speech is equal to the content embedding vector of the source speech and is defined as

$$\mathcal{L}_{con} = \mathbb{E}_{\mathbf{x}_s,\tilde{\mathbf{x}}} \left\| E_c(\mathbf{x}_s) - E_c(\tilde{\mathbf{x}}) \right\|_2^2. \tag{7}$$

KL loss KL loss [26, 27, 41] is a constraint that makes the distribution of the speaker embedding vector close to a normal distribution. The KL loss is defined as

$$\mathcal{L}_{kl} = \mathbb{E}_{\mathbf{x}_s}[\mathcal{D}_{KL}(p(\mathbf{z}_s|\mathbf{x}_s)||\mathcal{N}(\mathbf{z}|\mathbf{0}, \boldsymbol{I}))], \tag{8}$$

where $\mathcal{D}_{KL}(\cdot||\cdot)$ denotes KL divergence, and $p(\mathbf{z}_s|\mathbf{x}_s)$ indicates the output distribution of $E_s(\mathbf{x}_s)$. By constraining the speaker's latent space to the normal distribution, the speaker encoder makes generalizations to unseen speakers.

Full objective The full generator loss function can be summarized as follows

$$\mathcal{L}(E_c, E_s, G) = \lambda_{adv}\mathcal{L}_{adv} + \lambda_{asr}\mathcal{L}_{asr} + \lambda_{f0}\mathcal{L}_{f0} + \lambda_{rec}\mathcal{L}_{rec} + \lambda_{con}\mathcal{L}_{con} + \lambda_{kl}\mathcal{L}_{kl}, \tag{9}$$

where λ_{adv}, λ_{asr}, λ_{f0}, λ_{rec}, λ_{con}, and λ_{kl} are hyperparmeters for each loss. In addition, the discriminators are trained via only adversarial loss $\mathcal{L}_{adv}$.

4 Experiments

4.1 Datasets

For a fair performance comparison, the baseline and our proposed methods are trained with the VCTK dataset [42], with 44 h of utterances of 109 speakers. As in NVC-Net [37], six speakers are separated into unseen speakers. 90% and 10% of utterances of the remaining 103 speakers are randomly partitioned into a training set and a test set.

4.2 Implementation Details

For training, all datasets are downsampled to 24 kHz and randomly clipped to 38,540 samples (approximately 1.5 s) every epoch, and random clipping and random scaling are employed as the data augmentation. We train for a total of 500 epochs using the Adam optimizer with $\beta_1 = 0.5$, $\beta_2 = 0.9$, and a learning rate of 0.0001. The hyperparameters of the full loss are set to $\lambda_{adv} = 1$, $\lambda_{asr} = 5$, $\lambda_{f0} = 2.5$, $\lambda_{rec} = 10$, $\lambda_{con} = 10$, and $\lambda_{kl} = 0.02$ as mentioned in NVC-Net. The pre-trained networks mentioned in StarGANv2-VC[2] [17] are employed as the F0 extraction network and the ASR network, which are pre-trained with fundamental frequency given by World vocoder [43] and 24kHz phoneme level data, respectively.

AdaIN-VC [7], Again-VC [33], VQMIVC [11], NVC-Net [37], and TriAAN [44] are employed as the baseline methods to compare the performance of WaveVC. AdaIN-VC,[3] Again-VC,[4] VQMIVC,[5] and TriAAN-VC[6] are trained with the same dataset mentioned

[2] https://github.com/yl4579/StarGANv2-VC.

[3] https://github.com/jjery2243542/adaptive_voice_conversion.

[4] https://github.com/KimythAnly/AGAIN-VC.

[5] https://github.com/Wendison/VQMIVC.

[6] https://github.com/winddori2002/TriAAN-VC.

Table 1 The objective quality assessments on VC methods

Method	Seen to seen			Unseen to unseen		
	pMOS	CER (%)	WER (%)	pMOS	CER (%)	WER (%)
Source speech	3.01	1.6	5.9	3.03	1.9	5.5
AdaIN-VC	2.75	13.6	27.1	2.79	12.2	23.6
Again-VC	2.55	27.0	45.3	2.41	29.0	48.1
VQMIVC	2.68	10.4	21.8	2.64	11.5	22.5
NVC-Net	2.87	13.3	24.5	2.79	10.9	21.4
TriAAN-VC	2.70	10.8	21.0	2.59	12.9	24.7
WaveVC	2.90	5.4	12.7	2.84	4.7	10.9

in Sect. 4.1 by using the official code on the website. Unlike WaveVC and other baseline methods, VQMIVC and TriAAN-VC are experimented with by downsampling the dataset to 16kHz, while Again-VC is experimented with by downsampling the dataset to 22050Hz as mentioned in the references. NVC-Net[7] is reconfigured and trained with PyTorch [45].

4.3 Objective Quality Assessment

For the objective quality assessment, 600 samples were randomly generated from seen-to-seen and unseen-to-unseen cases, respectively. MBNet[8] [46]-based predicted mean opinion score (pMOS) evaluation, Wav2Vec2.0 [47]-based character error rate (CER), and word error rate (WER) are performed on the sampled data for the objective quality assessments. Wav2Vec2.0 uses self-supervised learning with unlabeled data for diverse quantized representation and is fine-tuned with labeled data by using connectionist temporal classification (CTC) loss [48]. The objective quality assessments are summarized in Table 1. The first row shows the results for the source speech used as the input. According to the experimental results, WaveVC outperforms other baseline methods in objective quality assessments. In the seen-to-seen case, WaveVC achieves 5.4% CER and 12.7% WER, and in the unseen-to-unseen case, it shows 4.7% CER and 10.9% WER. In particular, the CER of WaveVC is close to half of the next lowest-performing method. The WER of WaveVC is also significantly lower than other baseline methods. Meanwhile, WaveVC and NVC-Net show higher pMOS than other VC methods using the vocoder. Methods that directly synthesize raw audio waveforms show high audio quality because they minimize information loss that occurs during the conversion process to mel-spectrogram. Consequently, these results mean that WaveVC not only generates high-quality speech but also preserves the utterance information of the source speech well.

4.4 Subjective Quality Assessment

The mean opinion score (MOS) is conducted on naturalness and similarity metrics to evaluate VC performance (Table 2). The naturalness metric is scored from 1 to 5 by evaluating whether the converted speech has noise and distortion. The similarity metric is scored from 1 to 5 on

[7] https://github.com/kyungdeuk/NVCNet-pytorch.

[8] https://github.com/sky1456723/Pytorch-MBNet.

Table 2 The experimental results on ablation study according to losses

Method	Seen to seen			Unseen to unseen		
	pMOS	CER (%)	WER (%)	pMOS	CER (%)	WER (%)
WaveVC	2.90	5.4	12.7	2.84	4.7	10.9
WaveVC w/o speech loss	3.04	13.7	26.2	2.87	14.9	27.3
WaveVC w/o F0 loss	2.80	5.3	12.3	2.61	4.8	11.2

Table 3 The MOS results on VC methods

Metric	Method	Seen to seen				Unseen to unseen			
		M2M	M2F	F2M	F2F	M2M	M2F	F2M	F2F
Naturalness	Ground truth	4.28				4.26			
	AdaIN-VC	1.16	1.24	1.30	1.32	1.98	1.78	1.56	1.56
	Again-VC	2.14	1.56	1.98	2.00	2.16	1.96	1.60	1.94
	NVC-Net	3.06	2.78	3.44	3.50	3.54	3.46	3.08	3.32
	WaveVC	3.86	3.50	4.24	4.26	3.98	4.22	3.78	4.12
Similarity	AdaIN-VC	1.56	1.64	1.60	1.44	1.74	1.70	1.50	1.56
	Again-VC	2.38	1.84	2.08	2.04	1.98	1.86	1.70	2.02
	NVC-Net	3.14	3.14	3.46	3.54	3.16	3.44	3.26	3.30
	WaveVC	3.70	3.76	4.14	4.00	3.50	4.00	3.70	3.78

how similar the converted voice is to the target speaker. The MOS evaluation is performed on 20 samples, each in seen-to-seen and unseen-to-unseen cases, by a total of 20 participants. The MOS results are summarized in Table 3.

AdaIN-VC and Again-VC do not convert well and sometimes fail to generate data. A large number of speakers during training seems impossible to cover with the zero-shot learning-based VC methods. The adversarial raw audio VC methods show relatively higher MOS values than the zero-share learning-based methods. WaveVC shows from 0.72 to 0.8 higher naturalness score and from 0.46 to 0.68 higher similarity score than NVC-Net in the seen-to-seen case. In particular, in the case of WaveVC's F2M and F2F, naturalness scores similar to the ground truth are shown. In the unseen-to-unseen case, WaveVC gets from 0.44 to 0.80 higher naturalness score and from 0.34 to 0.56 higher similarity score than NVC-Net. These scores indicate that WaveVC performs speech and fundamental frequency consistent VC. As a result, WaveVC not only performs adversarial raw audio VC but also improves performance by concatenating the fundamental frequency feature into the content embedding vector and applying the speech loss and the F0 loss.

4.5 Ablation Study

The ablation study is performed to compare how the speech loss and the F0 loss affect the converted speech. When speech loss is not applied, CER and WER increase significantly. These results indicate that the speech loss helps preserve the source speech's utterance information well. On the other hand, the pMOS values decrease when using not the F0 loss. It

Table 4 The experimental results on ablation study for speaker verification

Method	Cosine similarity	SV resemblyzer (%)	SV TitaNet (%)
WaveVC	0.731	75.0	95.3
AdaIN-VC	0.646	30.5	25.8
Again-VC	0.690	51.8	12.8
VQMIVC	0.770	89.3	97.7
TriAAN-VC	0.789	95.7	99.0

converts into high-quality speech while preserving the information about the fundamental frequency of the source speaker and injecting the information at the time of conversion.

Additionally, cosine similarity and speaker verification (SV) accuracy are measured by using the pre-trained speaker verification model such as Resemblyzer[9] and TitaNet[10] [49] (Table 4). Resemblyzer is composed of LSTM and is trained by using the generalized end-to-end (GE2E) loss [50]. TitaNet is based on ContextNet ASR architecture [51] and is trained by using additive angular margin (AAM) loss [52] The cosine similarity is calculated between the converted speech's embedding vector and the seen target speech's embedding vector by using Resemblyzer. The threshold for SV using Resemblyzer is set to the equal error rate of the VCTK dataset as mentioned in TraiAAN-VC [44]. The autoencoder-based methods such as AdaIN-VC and Again-VC show significantly low cosine similarity and SV accuracy. These results indicate that they have a limitation in disentangling the content and speaker information. To solve this problem, VQMIVC uses VQCPC [47, 53], and TriAAN-VC employs an attention-based mechanism, time-wise instance normalization, and CPC. Meanwhile, WaveVC archives higher cosine similarity and SV accuracy than AdaIN-VC and Again-VC but shows lower performances than VQMIVC and TriAAN-VC. We can consider two reasons for these results. One is that CPC [54] is employed for extracting only content information from source speech. Because CPC is trained to predict future contextual information using current features, it is advantageous for extracting content information regardless of the speaker. The other is to use an attention mechanism to inject the target speaker's information into the content information. To overcome this limitation in future works, we will apply the VQCPC-based method to the content encoder and the attention-based mechanism to the speaker encoder.

5 Conclusions

In this paper, we proposed the adversarial raw audio VC method called WaveVC, which does not require a separate vocoder because it performs VC directly on raw audio. In addition, the proposed WaveVC performed speech and fundamental frequency consistent VC by reflecting the fundamental frequency information to the content embedding vector and adding two losses: speech loss and F0 loss. To compare the performance of WaveVC with other VC methods, we conducted a MOS evaluation for the naturalness and similarity of the VC results. As a result, WaveVC not only produced better performance than other competing VC methods but also showed a level of naturalness similar to the ground truth. In addition, in

[9] https://github.com/resemble-ai/Resemblyzer.

[10] https://huggingface.co/nvidia/speakerverification_en_titanet_large.

objective quality assessments such as pMOS, CER, and WER, WaveVC showed significantly better performance than other VC methods.

Acknowledgements This material is based upon work supported by the Air Force Office of Scientific Research under award number FA2386-23-1-4098.

References

1. Li Y, Jin Y, Kwak J, Yoon D, Han D, Ko H (2021) Adaptive content feature enhancement gan for multimodal selfie to anime translation
2. Ahn S, Ko H (2005) Background noise reduction via dual-channel scheme for speech recognition in vehicular environment. IEEE Trans Consum Electron 51(1):22–27
3. Kim G, Han DK, Ko H (2021) Specmix: a mixed sample data augmentation method for training with time-frequency domain features. arXiv preprint arXiv:2108.03020
4. Nachmani E, Wolf L (2019) Unsupervised singing voice conversion. arXiv preprint arXiv:1904.06590
5. Nakamura K, Toda T, Saruwatari H, Shikano K (2012) Speaking-aid systems using gmm-based voice conversion for electrolaryngeal speech. Speech Commun 54(1):134–146
6. Qian K, Zhang Y, Chang S, Yang X, Hasegawa-Johnson M (2019) Autovc: zero-shot voice style transfer with only autoencoder loss. In: International conference on machine learning. PMLR, pp 5210–5219
7. Chou J, Yeh C, Lee H (2019) One-shot voice conversion by separating speaker and content representations with instance normalization. arXiv preprint arXiv:1904.05742
8. Qian K, Jin Z, Hasegawa-Johnson M, Mysore GJ (2020) F0-consistent many-to-many non-parallel voice conversion via conditional autoencoder. In: ICASSP 2020—2020 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, pp 6284–6288
9. Wu D-Y, Lee H (2020) One-shot voice conversion by vector quantization. In: ICASSP 2020—2020 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, pp 7734–7738
10. Wu D-Y, Chen Y-H, Lee H-Y (2020) Vqvc+: one-shot voice conversion by vector quantization and u-net architecture. arXiv preprint arXiv:2006.04154
11. Wang D, Deng L, Yeung YT, Chen X, Liu X, Meng H (2021) Vqmivc: vector quantization and mutual information-based unsupervised speech representation disentanglement for one-shot voice conversion. arXiv preprint arXiv:2106.10132
12. Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y (2020) Generative adversarial networks. Commun ACM 63(11):139–144
13. Mun S, Park S, Han DK, Ko H (2017) Generative adversarial network based acoustic scene training set augmentation and selection using svm hyper-plane. In: DCASE, pp 93–102
14. Choi Y, Choi M, Kim M, Ha J-W, Kim S, Choo J (2018) Stargan: unified generative adversarial networks for multi-domain image-to-image translation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 8789–8797
15. Kameoka H, Kaneko T, Tanaka K, Hojo N (2018) Stargan-vc: non-parallel many-to-many voice conversion using star generative adversarial networks. In: 2018 IEEE spoken language technology workshop (SLT). IEEE, pp 266–273
16. Kaneko T, Kameoka H, Tanaka K, Hojo N (2019) Stargan-vc2: rethinking conditional methods for stargan-based voice conversion. arXiv preprint arXiv:1907.12279
17. Li YA, Zare A, Mesgarani N (2021) Starganv2-vc: a diverse, unsupervised, non-parallel framework for natural-sounding voice conversion. arXiv preprint arXiv:2107.10394
18. Kumar K, Kumar R, Boissiere T, Gestin L, Teoh WZ, Sotelo J, Brébisson A, Bengio Y, Courville AC (2019) Melgan: generative adversarial networks for conditional waveform synthesis. Adv Neural Inf Process Syst 32:66

19. Yamamoto R, Song E, Kim J-M (2020) Parallel wavegan: a fast waveform generation model based on generative adversarial networks with multi-resolution spectrogram. In: ICASSP 2020—2020 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, pp 6199–6203
20. Kong J, Kim J, Bae J (2020) Hifi-gan: generative adversarial networks for efficient and high fidelity speech synthesis. Adv Neural Inf Process Syst 33:17022–17033
21. Wu Y-C, Kobayashi K, Hayashi T, Tobing PL, Toda T (2018) Collapsed speech segment detection and suppression for wavenet vocoder. arXiv preprint arXiv:1804.11055
22. Oord Avd, Dieleman S, Zen H, Simonyan K, Vinyals O, Graves A, Kalchbrenner N, Senior A, Kavukcuoglu K (2016) Wavenet: a generative model for raw audio. arXiv preprint arXiv:1609.03499
23. Arık SÖ, Chrzanowski M, Coates A, Diamos G, Gibiansky A, Kang Y, Li X, Miller J, Ng A, Raiman J (2017) Deep voice: real-time neural text-to-speech. In: International conference on machine learning. PMLR, pp 195–204
24. Wang Y, Skerry-Ryan R, Stanton D, Wu Y, Weiss RJ, Jaitly N, Yang Z, Xiao Y, Chen Z, Bengio S et al (2017) Tacotron: towards end-to-end speech synthesis. arXiv preprint arXiv:1703.10135
25. Gibiansky A, Arik S, Diamos G, Miller J, Peng K, Ping W, Raiman J, Zhou Y (2017) Deep voice 2: multi-speaker neural text-to-speech. Adv Neural Inf Process Syst 30:66
26. Zhang Y-J, Pan S, He L, Ling Z-H (2019) Learning latent representations for style control and transfer in end-to-end speech synthesis. In: ICASSP 2019—2019 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, pp 6945–6949
27. Kingma DP, Welling M (2013) Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114
28. Ping W, Peng K, Chen J (2018) Clarinet: parallel wave generation in end-to-end text-to-speech. arXiv preprint arXiv:1807.07281
29. Ren Y, Hu C, Tan X, Qin T, Zhao S, Zhao Z, Liu T-Y (2020) Fastspeech 2: fast and high-quality end-to-end text to speech. arXiv preprint arXiv:2006.04558
30. Donahue J, Dieleman S, Bińkowski M, Elsen E, Simonyan K (2020) End-to-end adversarial text-to-speech. arXiv preprint arXiv:2006.03575
31. Huang X, Belongie S (2017) Arbitrary style transfer in real-time with adaptive instance normalization. In: Proceedings of the IEEE international conference on computer vision, pp 1501–1510
32. Ko K, Lee B, Hong J, Han D, Ko H (2021) Deep degradation prior for real-world super-resolution. In: BMVC
33. Chen Y-H, Wu D-Y, Wu T-H, Lee H (2021) Again-vc: a one-shot voice conversion using activation guidance and adaptive instance normalization. In: ICASSP 2021—2021 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, pp 5954–5958
34. Lin YY, Chien C-M, Lin J-H, Lee H, Lee L (2021) Fragmentvc: any-to-any voice conversion by end-to-end extracting and fusing fine-grained voice fragments with attention. In: ICASSP 2021—2021 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, pp 5939–5943
35. Lee S-H, Kim J-H, Chung H, Lee S-W (2021) Voicemixer: adversarial voice style mixup. Adv Neural Inf Process Syst 34:294–308
36. Wang Q, Zhang X, Wang J, Cheng N, Xiao J (2022) Drvc: a framework of any-to-any voice conversion with self-supervised learning. In: ICASSP 2022—2022 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, pp 3184–3188
37. Nguyen B, Cardinaux F (2022) Nvc-net: end-to-end adversarial voice conversion. In: ICASSP 2022—2022 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, pp 7012–7016
38. Hendrycks D, Gimpel K (2016) Gaussian error linear units (gelus). arXiv preprint arXiv:1606.08415
39. Kum S, Nam J (2019) Joint detection and classification of singing voice melody using convolutional recurrent neural networks. Appl Sci 9(7):1324
40. Kim S, Hori T, Watanabe S (2017) Joint ctc-attention based end-to-end speech recognition using multi-task learning. In: 2017 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, pp 4835–4839
41. Lee B, Ko K, Hong J, Ku B, Ko H (2022) Information bottleneck measurement for compressed sensing image reconstruction. IEEE Signal Process Lett 29:1943–1947
42. Yamagishi J, Veaux C, MacDonald K (2019) CSTR VCTK Corpus: English multi-speaker corpus for CSTR voice cloning toolkit (version 0.92). University of Edinburgh. The Centre for Speech Technology Research (CSTR). https://doi.org/10.7488/ds/2645
43. Morise M, Yokomori F, Ozawa K (2016) World: a vocoder-based high-quality speech synthesis system for real-time applications. IEICE Trans Inf Syst 99(7):1877–1884
44. Park HJ, Yang SW, Kim JS, Shin W, Han SW (2023) Triaan-vc: triple adaptive attention normalization for any-to-any voice conversion. In: ICASSP 2023—2023 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, pp 1–5

45. Paszke A, Gross S, Chintala S, Chanan G, Yang E, DeVito Z, Lin Z, Desmaison A, Antiga L, Lerer A (2017) Automatic differentiation in pytorch
46. Leng Y, Tan X, Zhao S, Soong F, Li X-Y, Qin T (2021) Mbnet: Mos prediction for synthesized speech with mean-bias network. In: ICASSP 2021—2021 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, pp 391–395
47. Baevski A, Zhou Y, Mohamed A, Auli M (2020) wav2vec 2.0: a framework for self-supervised learning of speech representations. Adv Neural Inf Process Syst 33:12449–12460
48. Baevski A, Auli M, Mohamed A (2019) Effectiveness of self-supervised pre-training for speech recognition. arXiv preprint arXiv:1911.03912
49. Koluguri NR, Park T, Ginsburg B (2022) Titanet: neural model for speaker representation with 1d depth-wise separable convolutions and global context. In: ICASSP 2022—2022 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, pp 8102–8106
50. Wan L, Wang Q, Papir A, Moreno IL (2018) Generalized end-to-end loss for speaker verification. In: 2018 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, pp 4879–4883
51. Han W, Zhang Z, Zhang Y, Yu J, Chiu C-C, Qin J, Gulati A, Pang R, Wu Y (2020) Contextnet: improving convolutional neural networks for automatic speech recognition with global context. arXiv preprint arXiv:2005.03191
52. Deng J, Guo J, Xue N, Zafeiriou S (2019) Arcface: additive angular margin loss for deep face recognition. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 4690–4699
53. Van Niekerk B, Nortje L, Kamper H (2020) Vector-quantized neural networks for acoustic unit discovery in the zerospeech 2020 challenge. arXiv preprint arXiv:2005.09409
54. Oord Avd, Li Y, Vinyals O (2018) Representation learning with contrastive predictive coding. arXiv preprint arXiv:1807.03748