Any-to-Any Voice Conversion with Multi-layer Speaker Adaptation and Content Supervision

Xuexin Xu, Liang Shi, Xunquan Chen, Pingyuan Lin, Jie Lian, Jinhui Chen, Zhihong Zhang*, Edwin R. Hancock, *Fellow, IEEE*

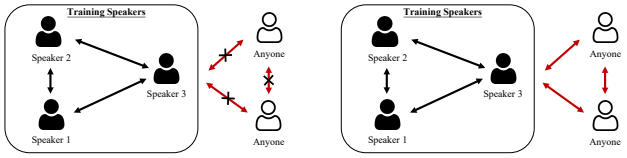
***Abstract*—Any-to-any voice conversion can be performed among arbitrary speakers, even with a single reference utterance. Many related studies have demonstrated that it can be effectively implemented by speech representation disentanglement. However, most existing solutions fuse the speaker representations into the content features globally without considering their distribution difference. Additionally, in the any-to-any scenario, there is no effective method ensuring the consistency of linguistic content without text transcription or additional information extracted from additional modules (e.g., automatic speech recognition). Hence, to alleviate the above problems, this paper proposes SACS-VC, a novel any-to-any voice conversion method that combines two principal modules: Speaker Adaptation and Content Supervision. Specifically, we rearrange the timbre representations according to the content distribution using a temporal attention mechanism to obtain finer-grained speaker timbre information for each content feature. Meanwhile, we associate the converted outputs and source utterances directly to supervise the consistency of the semantic content in an unsupervised manner. This is achieved using contrastive learning based on the corresponding and non-corresponding locations of content features. It should be noted that SACS-VC can be implemented using a non-parallel speech corpus without any pertaining. The experimental results demonstrate that the proposed method outperforms current state-of-the-art any-to-any voice conversion systems in objective and subjective evaluation settings.**

***Index Terms*—Voice conversion, attention mechanism, contrastive learning, feature disentanglement.**

I. Introduction

VOICE conversion (VC) converts speaker identity from a source utterance to that of a target speaker while preserving the original linguistic content. This approach is widely used in many applications, such as personalized speech synthesis and human–computer interaction.

Early work [1]–[6] focused relied on aligned parallel data, where any speech pairs from source and target speakers share the same linguistic content and are aligned in the temporal dimension. However, it is challenging to collect such data and time-consuming to align them. Besides, the restricted corpus availability limits the performance and generalizability of voice conversion. These limitations have motivated researchers to explore non-parallel voice conversion approaches [7]–[10], which led to a deep neural network that approximated a mapping function from the source speaker domain to the target speaker domain. For instance, CycleGAN-VC [9] and StarGAN-VC [8] employ cycle consistency to ensure that the invertible mapping results are identical to the source input. Although these methods attain an appealing performance without requiring a parallel corpus, they only involve a conversion process for predefined multiple speaker sets. As depicted in Fig. 1, when encountering arbitrary speakers, which may be unseen during training (outside the set of speakers used in training), the above VC methods have relatively limited conversion capabilities. To overcome such limitations, several any-to-any voice conversion methods have been explored [11]–[13]. Most of the existing any-to-any VC approaches are based on speech representation disentanglement. This effectively addresses the any-to-any conversion problem by decomposing speech into speaker timbre and linguistic content representations. Then, the speaker identity can be converted by only replacing the speaker timbre representation from one speaker to another. Fig. 2 illustrated this process. Many techniques have been proposed to separate speaker timbre information from linguistic content as much as possible. These include the information constrained bottleneck layer [12], [14], phoneme transcription guidance [15], vector quantization [13], [16], [17], normalization techniques [11], [18] and self-supervised speech representation [19]–[21].



(a) Many-to-many VC (b) Any-to-any VC

Fig. 1. Comparison between many-to-many and any-to-any voice conversion. The arrows represent voice conversion among different speakers.

However, most of these methods only embed the speaker timbre without considering its relevance to content, which is an average global speaker feature. Nevertheless, using averaged timbre information sacrifices the timbre modeling capability

*Corresponding author: Zhihong Zhang (E-mail: zhihong@xmu.edu.cn)

Xuexin Xu, Liang Shi, Pingyuan Lin, Jie Lian and Zhihong Zhang are with the School of Informatics, Xiamen University, Xiamen, China.

Xunquan Chen is with the Graduate School of System Informatics, Kobe University, Kobe, Japan.

Jinhui Chen is with the Faculty of Systems Engineering, Wakayama University, Wakayama, Japan.

Edwin R. Hancock is with the Department of Computer Science, The University of York, York, UK.

This work is supported by the National Natural Science Foundation of China under Grant (62176227, U2066213), Fundamental Research Funds for the Central Universities (20720210047), JSPS KAKENHI under Grant (19H00597), and the Research Support Fund of Wakayama University.

of local phonemics and processes all local content features using the same transformation function in voice conversion. For example, the pioneering work reported in [11] proposed a simple yet effective method that applies the global mean together with the variance of the target speech to the source utterance in a deep feature space. Since the required statistics are calculated globally from a fixed-length speaker representation, the fine-grained details and phoneme-wise patterns are largely discarded. Furthermore, silence segments affect the speaker's representation because they contain almost no useful information. The same issue exists in AutoVC [12], which applies a pre-trained speaker encoder to extract the global speaker timbre representations. To obtain the fine-grained speaker embedding for each content representation scale, typically, more attention should be paid to the most similar phonemic pronunciation of the target utterances, and then the corresponding timbre representation should be extracted and embedded for these temporal locations.

Unfortunately, once the network probes the local fine-grained speaker information, unreliable information may contaminate the corresponding content due to incomplete decoupling of speech representations. The residual mutual information between them at the same locations will restrain the original features. Therefore, the linguistic content of the converted speech is usually distorted or ambiguous, which is unacceptable for VC. In fact, existing state-of-art any-to-any VC methods, such as AdaIN-VC [11] and AutoVC [12], are devoted to achieving arbitrary timbre transfer without a parallel aligned corpus. Hence, they all fail to achieve effective supervision concerning the linguistic content without any additional processing modules (*e.g.,* text transcription and automatic speech recognition). This unsupervised learning framework only includes the main objective of reconstructing input utterances (as shown in Fig. 2). Specifically, a pioneering study proposed CVC [22] which preserves content information by contrastive learning but can only perform VC with many-to-one or any-to-one mapping, limiting the flexibility of VC in the real world. Therefore, the ultimate goal of VC can be defined in a more detailed manner as transforming the speaker timbre as much as possible without losing semantic content.

This paper addresses these problems and better balances transferring speaker timbre and preserving semantic content. Therefore, we propose a novel any-to-any VC framework called ***SACS-VC***, which introduces ***S**peaker **A**daptation* and ***C**ontent **S**upervision* to resolve the above problems. Speaker adaptation can adaptively rearrange the speaker timbre representations according to the content distribution using a temporal attention mechanism and then perform timbre transfer on each content feature. Content supervision is self-supervised to preserve semantic content directly.

Specifically, in SACS-VC, the temporal attention map is learned jointly from the content and speaker features by implicitly aligning similar phonemic pronunciations. Subsequently, the speaker features are rearranged concerning this map, and then the stylized features are generated by the position-wise addition of rearranged speaker features to give content features. Motivated by previous research [23], we realize the content supervision by associating the converted speech and source input directly using contrastive learning. Specifically, we maximize the mutual information of the semantic content between the converted and source speech and consider the distance error of feature value space between them. Following the guidance of mutual information and feature space distance. Although we only consider a non-parallel speech corpus in the training stage, we establish the semantic correspondences between the source input and converted output based on content features and ensure that the semantic content is preserved as much as possible during the entire conversion process. To some extent, preserving the linguistic content helps to decouple the speech representations better. Meanwhile, we encapsulate the whole framework in an adversarial training strategy to enhance the synthetic speech quality using a U-Net-like [24] multi-scale architecture. Considering the different temporal scales in audio features, the above operations (*i.e.*, speaker adaptation, and content supervision) can consider different feature scales of the deep embedding. Note that *SACS-VC* can achieve a more fine-grained speaker timbre transformation for each phonemic and preserve semantic content consistency as much as possible during VC. Our main contributions can be summarized as follows:

- We propose a *speaker adaptation* module to adaptively rearrange the speaker timbre distribution according to the content distribution using a temporal attention mechanism. In this way, we generate the corresponding speaker features for each content feature providing a more fine-grained and appropriate timbre pattern that depends on semantic content.
- A novel optimization objective referred to as *content supervision* is proposed. These associates converted outputs and source utterances and helped the method to preserve the semantic content during VC by maximizing the mutual information between them.
- We consider both high and low-level deep features at different temporal scales to achieve better convergence. Additionally, an adversarial strategy and multi-scale architecture are adopted to enhance the quality of the audio signals generated. Both subjective and objective experimental results demonstrate that our method is better than or comparable to existing state-of-the-art any-to-any VC methods on real-world VCTK [25] datasets.

The remainder of this paper is organized as follows. Sec. II briefly surveys the related literature. Secs. III and IV present our SACS-VC method and Sec. V reports our experimental results. Finally, Sec. VI concludes this paper and suggests future research directions.

II. Related work

Non-parallel VC is an unsupervised learning process, and its learning difficulty lies in constructing a mapping relationship between non-parallel speech corpus. Cascade VC models [26], [27] enable VC by extracting linguistic content through an Automatic Speech Recognition (ASR) model and then feeding it into a Text-To-Speech (TTS) model. Due to the speaker-independent property of Phonetic PosterioGram

(PPG), which can be extracted from a pre-trained ASR model, it has succeeded in VC [28]. However, the ASR model's performance limits the conversion quality of these methods, and a large amount of data is required to pre-train the ASR system, restricting VC's flexible application. Besides, deep generative models bring new opportunities to VC, including VAEs [29], GANs [30], and DPMs [31]. According to the different frameworks, we divide non-parallel VC into direct transformation- and feature disentanglement-based

1) Direct transformation-based VC: Many researchers have developed feed-forward-based networks to achieve a direct transformation from one speaker to another to remove the requirement of a parallel corpus without additional data or pre-trained models. Some work [9], [32]–[34] use non-parallel VC networks, which can only achieve one-to-one conversion by training an independent network. VC among multiple speakers is a key enabling technology for various applications. Therefore, Kameoka *et al.* extended an image-to-image translation method StarGAN [35] to develop StarGAN-VC [8]. Chou *et al.* [7] employed a two-stage training strategy and adversarial speaker classifier to remove further speaker-dependent information from linguistic representations. Lee *et al.* [36] overcame the drawbacks of CycleGAN-based methods [9], [37] by conditioning the network on the speaker and performed many-to-many VC using a single network. Furthermore, CVC [22] adopts contrastive learning to replace the cycle-consistency mapping and allows better preservation of content information.

However, the above VC methods cannot efficiently transfer the speakers that are not present in the training data, i.e., they cannot model unseen data.

2) Feature disentanglement-based VC: Several studies based on speech representation disentanglement have attempted to decompose speech into speaker and content representations. These methods can achieve any-to-any VC by simply replacing the speaker representation. For instance, Qian *et al.* proposed AutoVC [12], which uses a pre-trained speaker encoder and imposes a restriction on the length of the bottleneck layer. In their subsequent work [14], they considered different properties of speech. Zhang *et al.* [15] used the corresponding phoneme transcriptions to guide the extraction of linguistic representations. Besides, Vector Quantization (VQ) was employed in [16] and [13] to separate the speaker-independent features. AdaIN-VC [11] demonstrated that instance normalization could effectively remove speaker style information and then applied adaptive instance normalization [38] to adjust global statistics (*i.e.*, mean and variance). Ishihara *et al.* [39] generated content-dependent speaker information using an attention mechanism, while in [40], the local and global timbre information was considered simultaneously. Self-supervised speech representations have also been employed for VC [19]–[21]. Indeed, Wang *et al.* [17] used mutual information to measure the dependencies between speech representations. Lei *et al.* [41] implemented a unified framework to achieve zero-shot text-to-speech and any-to-any VC simultaneously. Popov *et al.* [42] applied a diffusion model to VC, where the converted speech is synthesized by integrating the reference speaker information into the averaged speaker-independent speech representation in the reverse diffusion stage.

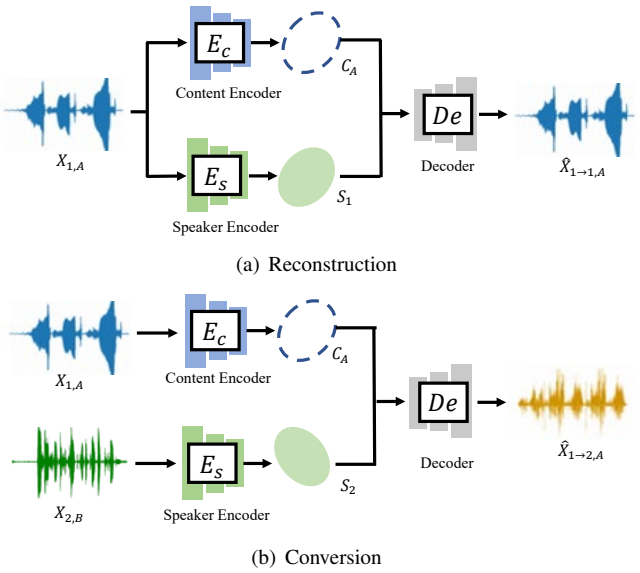


Fig. 2. Learning process of feature disentanglement-based VC.

Nevertheless, existing disentanglement VC methods only consider the reconstruction objective in the training procedure. However, preserving the semantic content during the conversion process is challenging, especially when using non-parallel data. Meanwhile, many previous studies only embedded the speaker representation into a predefined fixed-length vector, which is unsuitable for variable phonemic content. These methods fuse deep speaker features into the content features without considering the differences between feature distributions. To alleviate these problems, this study explores a better trade-off between transferring speaker timbre and preserving semantic content. Specifically, we design a speaker adaptation module to rearrange the speaker distribution by considering the details of the content distribution. This ensures that the embedded speaker representation is suited to the semantic content. Moreover, we propose a novel learning objective that uses contrastive learning to avoid semantic content changes during the conversion stage.

III. Model Framework of SACS-VC

A. Preliminaries

As illustrated in Fig. 2, AdaIN-VC [11] and AutoVC [12] disentangle content and speaker information from speech and transfer the target timbre by replacing the speaker representation. Their simple autoencoder framework comprises three modules: content encoder $E_c(\cdot)$, speaker encoder $E_s(\cdot)$, and decoder $De(\cdot,\cdot)$. In the training stage, the model only requires self-reconstruction from an input utterance to disentangle speech representations, which can be written as follows:

$$C_A = E_c(X_{1,A}),\ S_1 = E_s(X_{1,A}),\ \hat{X}_{1\to1,A} = De(C_A, S_1) \tag{1}$$

where $X_{1,A}$ denotes the utterance "A" produced by speaker "1", C_A is the linguistic information relevant to content "A"

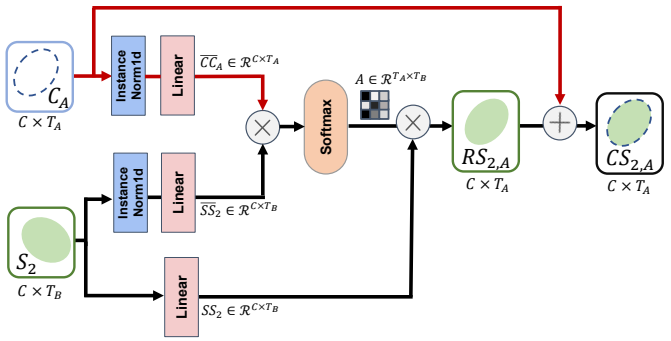


Fig. 3. Speaker Adaptation module. We rearrange the speaker distribution according to the content information and then fuse the content-dependent speaker features $RS_{2,A}$ into the content features C_A through point-wise addition to generate the stylized features $CS_{2,A}$.

captured from $E_c(\cdot)$, and S_1 indicates that speaker information about identity "1" is generated by the speaker encoder $E_s(\cdot)$. The Decoder $De(\cdot,\cdot)$ takes the content and speaker feature as inputs to synthesize the reconstructed utterances $\hat{X}_{1\to1,A}$.

To confine our attention to the non-parallel any-to-any VC setting, we require another reference speech to perform VC. Therefore, for a source speech $X_{1,A} \in \mathcal{R}^{C\times T_A}$ and reference speech $X_{2,B} \in \mathcal{R}^{C\times T_B}$, T_A and T_B denote the time lengths of the respective speeches depending on the utterance. The conversion process should transfer the speaker identity from "1" to "2" while preserving the source content "A", which can be written as

$$C_A = E_c(X_{1,A}),\ S_2 = E_s(X_{2,B}),\ \hat{X}_{1\to2,A} = De(C_A, S_2) \tag{2}$$

Based on the above process, we synthesize the converted speech $\hat{X}_{1\to2,A} \in \mathcal{R}^{C\times T_A}$ by replacing the speaker identity information from S_1 to S_2. However, this unsupervised learning process will inevitably lead the converted speech to miss some content information.

B. Speaker adaptation

According to Eq.(2), we generate the content features C_A and speaker representations S_2 from the source and reference speech, respectively. To overcome the negative effects of residual correlation information between C_A and S_2, the speaker adaptation module (SA) rearranges S_2 based on the content representations C_A and then generates content-dependent stylized features $RS_{2,A}$.

The SA module is illustrated in Figure 3. Initially, given a content feature $C_A \in \mathcal{R}^{C\times T_A}$, we conduct mean-variance channel-wise normalization to remove the timbre information [11] and then transform it linearly to generate the normalized feature $\overline{CC}_A$. We process the speaker features $S_2 \in \mathcal{R}^{C\times T_B}$ similarly to obtain the normalized speaker representation $\overline{SS}_2$. Meanwhile, we feed the speaker features S_2 into an additional linear layer, denoted by SS_2, but there is no normalization operation in this case. Similarly to the cross-attention operation, we first calculate the correlation matrix $A \in \mathcal{R}^{T_A\times T_B}$, which can be formulated as

$$A = SoftMax(\overline{CC}_A^T \otimes \overline{SS}_2) \tag{3}$$

where the dot-product measures the similarity between the two representations, and the position (i, j) of the correlation matrix A is used to measure the relation between the i^{th} content feature and j^{th} speaker feature. Then, we rearrange the speaker features SS_2 by taking the product of the correlation matrix A and SS_2 and appropriately generate the rearranged speaker feature $RS_{2,A} \in \mathcal{R}^{C\times T_A}$, expressed as follows:

$$RS_{2,A} = SS_2 \otimes A^T \tag{4}$$

In simple terms, for each position of the content feature, we automatically enumerate all positions of the speaker feature to align with the most similar phonemic position. Finally, we fuse the rearranged features into the content features to achieve VC as follows:

$$CS_{2,A} = RS_{2,A} + C_A \tag{5}$$

Through the above SA process, according to the content phonemic information, we generate a speaker feature consistent with the same distribution as the content feature that can easily be fused into the content features through feature addition to achieve fine-grained timbre construction of the target speaker. This fine-grained speaker representation automatically selects an appropriate speaking style for the semantic content information and avoids the interference caused by semantic inconsistencies to preserve the semantic content information of the source speech and improve the quality of VC.

C. Network architecture

The developed framework is based on a GAN [30], which typically comprises a generator and a discriminator. Given a non-parallel speech corpus, we sample two different speech instances $X_{1,A} \in \mathcal{R}^{C\times T_A}$ and $X_{2,B} \in \mathcal{R}^{C\times T_B}$, which come from two different speakers. The generator G is an auto-encoder framework that generates the converted speech $\hat{X}_{1\to2,B} = G(X_{1,A}, X_{2,B})$, which has similar content to $X_{1,A}$ and similar timbre to $X_{2,B}$. The discriminator constructs a weakly supervised learning strategy, distinguishing a real speech sample from a synthetic one while encouraging the generator to synthesize realistic speech of the target domain $X_{2,B}$. The network architectures are illustrated in Figs. 4 and 5.

1) Generator: The generator G can be divided into content encoder E_c, speaker encoder E_s, and decoder De. The generator comprises entirely convolution neural networks to achieve non-autoregressive generation. As depicted in Fig. 4, we capture the content speech features with different temporal scales in the content encoder and then restore them gradually in the decoder. This multi-scale architecture is very similar to U-Net [24].

In the encoders, we first employ the ConvBank layer [43], which stacks convolution layers with different kernel sizes to enlarge the receptive field and capture long-timescale information. Subsequently, several convolution layers are applied to generate high-level representations. The purely 1-dimensional convolution layers are implemented with a kernel size of 5, and the stride size depends on whether downsampling of the temporal scales is required. For the content encoder, we

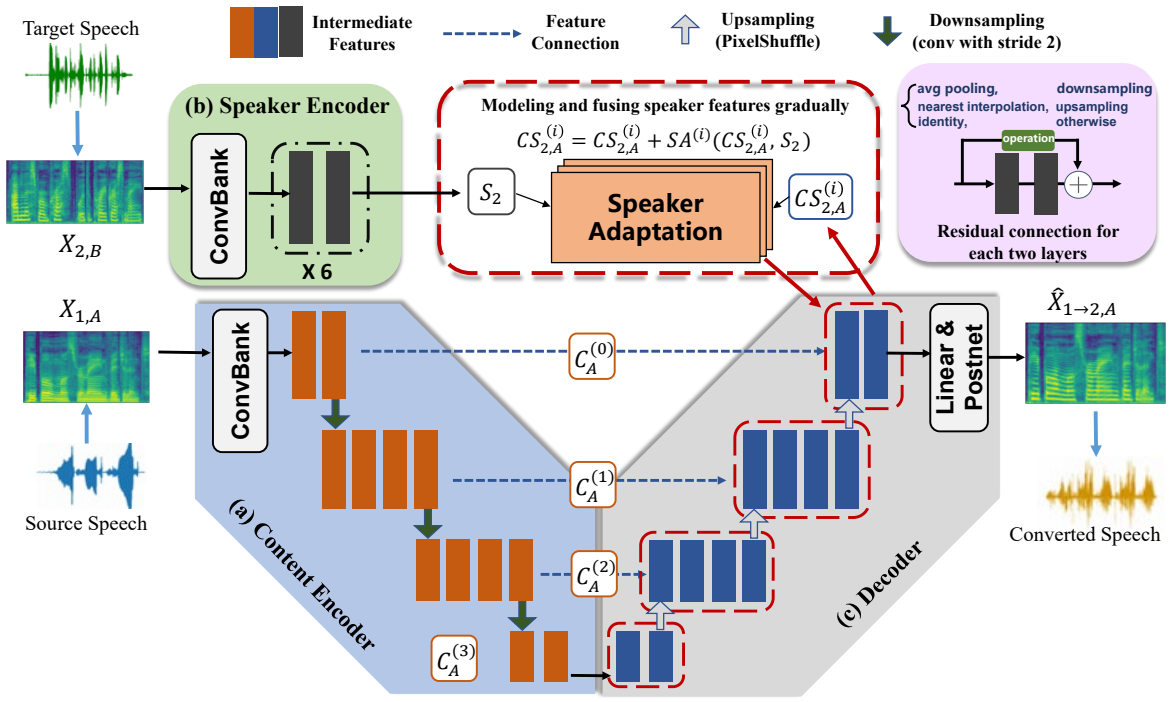


Fig. 4. Intuitive architecture diagram for the generator: (a) content encoder, (b) speaker encoder, and (c) decoder. A multi-scale architecture is implemented between (a) and (c). All features in the decoder (red dotted box) and the speaker features are fed into the speaker adaptation modules to generate stylized features by fusing the speaker information.

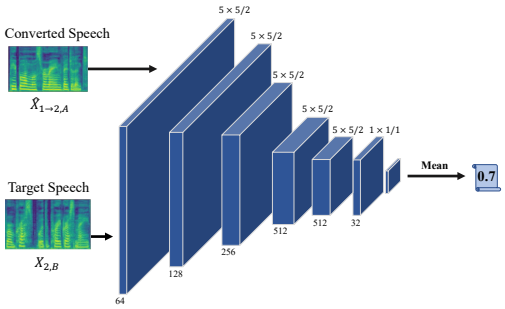


Fig. 5. Architecture diagram of the discriminator, composed of several 2d convolution layers.

downsample 3 times to decrease the feature resolution and adopt instance normalization after each convolution layer to eliminate the speaking timbre information [11]. Note that we do not downsample the temporal dimension in the speaker encoder, but the original temporal dimension is the same as the input acoustic features to preserve the overall information. To mitigate the training difficulties, we also implement *residual connections* [44] for each pair of convolution layers, except for the ConvBank layer. We also use average pooling to decrease the temporal resolution to match the feature shapes. As mentioned above, the content encoder will decrease the temporal scale gradually. Therefore, in addition to storing the output feature of the content encoder, we also store the intermediate features before each downsampling operation, *i.e.*, $C_A = \{C_A^{(0)}, C_A^{(1)}, C_A^{(2)}, C_A^{(3)}\}$, and the shapes of these features are $\{\mathcal{R}^{C\times T_A}, \mathcal{R}^{C\times \frac{T_A}{2}}, \mathcal{R}^{C\times \frac{T_A}{4}}, \mathcal{R}^{C\times \frac{T_A}{8}}\}$. The speaker encoder embeds $X_{2,B}$ to generate the speaker representation $S_2 \in \mathcal{R}^{C\times T_B}$ while preserving the temporal scale without any downsampling.

In the decoder, given the content features C_A and speaker feature S_2, there are two main basic operations: 1) restoring the temporal scale from the smallest-scale feature $C_A^{(3)}$ and 2) fusing the speaker feature S_2 into the content distribution using the speaker adaptation modules described in Sec. III-B. Specifically, we first initialize the feature $CS_{2,A}^{(i)}$ passed in the decoder as $C_A^{(3)}$. A set of convolution layers with a kernel size of 5 and stride of 1 are implemented in the decoder. To increase the temporal resolution, a PixelShuffle1d layer [45] is used for upsampling, and we use nearest neighbor interpolation so that the residual connections match the feature shape. We associate the feature map after upsampling and the corresponding content representation $C_A^{(i)}$ according to the same scale i, a skip-connection is implemented between $C_A^{(i)}$ and $CS_{2,A}^{(i)}$. To achieve fine-grained timbre modeling, we feed the restored and speaker features into the SA module, which can automatically adapt and fuse the speaker information into the converted feature according to the semantic correlation. This can be expressed as follows:

$$CS_{2,A}^{(i)} = CS_{2,A}^{(i)} + SA^{(i)}(CS_{2,A}^{(i)}, S_2) \tag{6}$$

To synthesize the converted speech, a pipeline is constructed using several consecutive "1)-2)" operations to restore the temporal scale of the features and then gradually fuse the speaker information. Then, we use a linear transformation to modify the channel to match the acoustic features. Finally, the *post network* [46] is appended but without batch normalization.

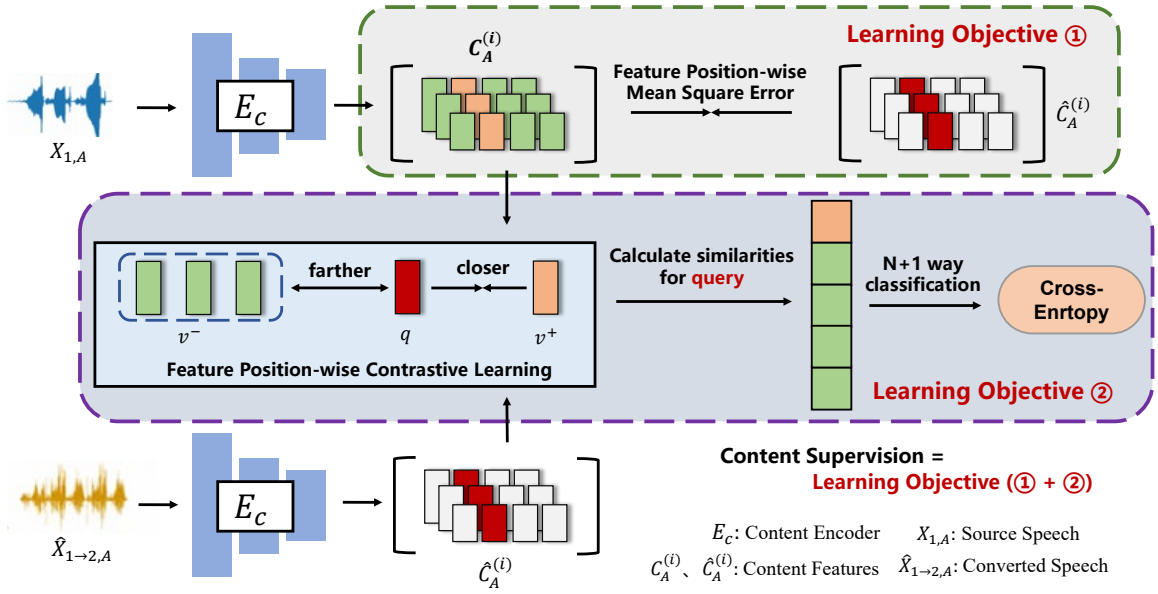


Fig. 6. Content supervision process flow. We establish the semantic content relationships between the source speech $X_{1,A}$ and converted speech $\hat{X}_{1\to2,A}$. Our content supervision has two learning objectives. First, we minimize the feature value errors between $X_{1,A}$ and $\hat{X}_{1\to2,A}$ at the same locations. Second, we maximize the mutual information between $X_{1,A}$, and $\hat{X}_{1\to2,A}$ using contrast learning while encouraging the content encoder E_c to distinguish the phonemic content.

This predicts a residual, which is added to the prediction to improve the overall reconstruction. The *post network* involves five convolution layers that use a hyperbolic tangent activation function in all but the final layer. The channel dimension is set to 512 in the first four layers and is reduced to 80 in the final layer. A dropout layer with a rate of 0.5 is placed after each layer.

2) Discriminator: Unlike the generator, the discriminator is constructed with 2D convolution layers similar to [7], [8] to capture the acoustic texture better. Specifically, we first reshape the input speech from $\mathcal{R}^{C\times T}$ to $\mathcal{R}^{1\times C\times T}$. Subsequently, there are 5 convolution layers with a stride of 2 and a kernel size of 5×5 to downsample the feature map gradually. The corresponding number of filters are 64, 128, 256, 512, and 512. Additionally, a convolution layer with unit kernel size and stride is appended to decrease the feature channel from 512 to 32. Finally, the output layer measures the degree of verisimilitude of the speech in the target domain. Instance normalization [47] and Leaky ReLu activation [48] with slope 0.01 are applied after each convolution layer, except for the final output layer. The mean value of the final feature is the output of the discriminator, which represents the confidence value that the input speech is the real speech of the target speaker.

IV. Learning Strategies of SACS-VC

A. Content supervision

VC should fully preserve the semantic content of the source speech while transferring the target speaker's timbre. However, we cannot completely decouple the speech representations and ensure that they are independent because incorporating speaker information will somewhat distort the content distribution. This distorts and enhances the ambiguity of the converted speech. To alleviate this problem, we propose the content supervision learning process, as illustrated in Fig. 6.

Given the source speech $X_{1,A}$ and target speech $X_{2,B}$ originating from different speakers, we accomplish VC and generate the converted speech $\hat{X}_{1\to2,A}$ based on Eq. (2). In an ideal VC system, although $\hat{X}_{1\to2,A}$ and $X_{1,A}$ belong to different speakers, the semantic content should be consistent throughout the conversion process. Given that we train the content encoder E_c to capture the linguistic content information of speech, the semantic content can be readily represented by the content feature. According to our framework setting in Sec. III-C, there are 4 different feature scales in the content feature stack, where a smaller scale corresponds to a larger receptive field. Therefore, the intuitive idea is to constrain the content features to be the same at the corresponding positions, that is, constraint the distances of content features between $\hat{X}_{1\to2,A}$ and $X_{1,A}$:

$$\mathcal{L}_{content} = \frac{1}{L}\sum_{i=0}^{L-1} \|\hat{C}_A^{(i)} - C_A^{(i)}\|_2 \tag{7}$$

where $L=4$, and i denotes the i^{th} index of the content feature stack. $\hat{C}_A$ and C_A are the content features extracted from $\hat{X}_{1\to2,A}$ and $X_{1,A}$, respectively. We use the mean squared error (MSE) to define the perceptual content loss. Ideally, the above approach alleviates the problem of content distortion or obfuscation. However, the content encoder E_c may learn a trivial function (such as loss of ability to distinguish phonemic content) and output the approximate representation for different semantic content. This is because we want to train the whole VC system end-to-end, and we inevitably update the parameters of E_c according to the above loss. To avoid E_c

losing the ability to capture content diversity, it is necessary to add one additional requirement to make the objective multi-task.

Motivated by the unpaired image translation method based on contrastive learning in [23], we build another learning strategy between $\hat{C}_A$ and C_A, based on the hypothesis that the semantic labels are the same during the VC. This learning strategy is based on contrastive learning, which maximizes the mutual correspondence information based on the InfoNCE loss [49]. This strategy further constrains the semantic content to be similar to each other and forces the content feature to distinguish different phonemic content, thus avoiding degrading the content encoder.

The key idea of contrastive learning is to construct three different types of vectors: a) "query" vector q, b) "positive" vector v^+, and c) N "negative" samples v^-. These are the column vector sampled from C_A and $\hat{C}_A$ for all temporal positions T of content features. There is one positive sample and the remaining N negative samples (*i.e.*, $T = N + 1$). Thus, $v, v^+ \in \mathcal{R}^{C\times 1}$ and $v^- \in \mathcal{R}^{C\times N}$. In our context, a query refers to a certain column vector sampled from $\hat{C}_A$, v^+, which corresponds with the same position of q in C_A, and v^- are the remaining elements of the feature set in C_A. We want q and v^+ to be close and q and each item in v^- to be far away. This can enforce the content encoder to output a similar embedding at the same temporal position and generate distinguishable representations at distinct locations. This multi-objective optimization problem can be also viewed as a multi-classification problem with $N + 1$ classes, maximizing the probability of selecting a positive sample v^+ over all negatives v^- to achieve contrastive learning indirectly. Specifically, the cross-entropy loss will be calculated to maximize the mutual information. This is achieved by maximizing the probability of matching the positive sample with the query vector. Indeed, we normalize each of these three vectors using the $\mathbf{L}2$ norm, which is mathematically formulated as follows:

$$\ell(q, v^+, v^-) = -\log\left[\frac{\exp(\frac{q\cdot v^+}{\tau})}{\exp(\frac{q\cdot v^+}{\tau}) + \sum_{n=1}^{N}\exp(\frac{q\cdot v_n^-}{\tau})}\right] \quad (8)$$

where v_n^- denotes the n^{th} negative sample and τ is a temperature parameter used to scale the feature distances. We maximize the mutual information between C_A and $\hat{C}_A$ by minimizing the above learning objective.

Due to our multi-scale architecture, we expand Eq. (8) to all scales of content features. For any scale i of the content features, we first select the n^{th} column vector of $\hat{C}_A^{(i)} \in \mathcal{R}^{C\times T_A^{(i)}}$ as the query vector and create the corresponding positive vector v_i^n and negative vectors $v_i^{(N_A^{(i)}+1)\setminus n}$, where $T_A^{(i)} = N_A^{(i)} + 1$. Subsequently, we build contrastive learning by enumerating all locations of content features as query vectors at each scale. As a result, the second objective can be expressed as

$$\mathcal{L}_{contrast} = \frac{1}{L}\sum_{l=0}^{L-1}\frac{1}{N_A^{(i)}+1}\sum_{n=1}^{N_A^{(i)}+1}\ell\left(q_i^n, v_i^n, v_i^{(N_A^{(i)}+1)\setminus n}\right) \quad (9)$$

where $L = 4$ corresponds to the 4 items in the content feature stack, and $N_A^{(i)}$ depends on the temporal dimension of the content features at different scales.

Note that our model solely relies on a self-supervised learning strategy without additional modules, and by using the above two types of constraints (mutual information and value constraint), we ensure that the semantic content information is preserved as much as possible during the entire VC process. In this way, the content features of the converted output will be similar to the source input and distinguish it from alternative phonemic content.

B. Loss function

In a non-parallel VC scenario, the two arbitrary sampled speech instances $\{X_{1,A}, X_{2,B}\} \sim \mathcal{X}$ make up the inputs of SACS-VC. To translate the source speech to sound like the target speaker, our proposed network is optimized in the training stage through three types of loss functions, as illustrated in Fig. 7.

1) Reconstruction loss: The reconstruction loss assists the generator in preserving the consistency of the spectrogram when using the same speech sample for both the input content speech and input reference speech:

$$\mathcal{L}_{recon}(X_{1,A}, \hat{X}_{1\to 1,A}) = \mathbf{E}_{X_{1,A}\sim\mathcal{X}}\|\hat{X}_{1\to 1,A} - X_{1,A})\|_1 \quad (10)$$

where $\hat{X}_{1\to 1,A}$ is the self-reconstruction procedure in Eq. (1), and the $\mathcal{L}_1$ distance (norm) measures the differences between the source and the corresponding reconstructed input. This reconstruction loss ensures that the auto-encoder architecture does discard much information and encourages the model to synthesize clean and understandable speech. It is also an essential part and main objective for feature disentanglement-based any-to-any VC methods [11], [12], [20].

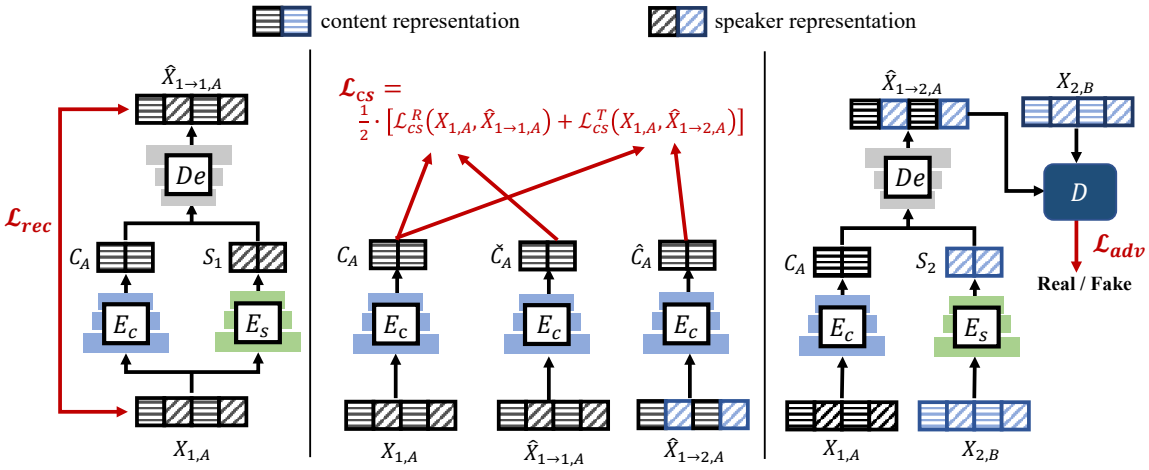
2) Content supervision loss: As discussed in Sec. IV-A, we use two different learning objectives to preserve the consistency of the semantic content during the VC process. Thus, the content supervision loss depends on Eqs.(7) and (9) in weighted combination:

$$\mathcal{L}_{cs}^T(X_{1,A}, \hat{X}_{1\to 2,A}) = \mathbf{E}_{\{X_{1,A}, X_{2,B}\}\sim\mathcal{X}}\, c_1\cdot\mathcal{L}_{content} + \mathcal{L}_{contrast} \quad (11)$$

where the coefficient c_1 is set to 0.5 to determine the relative weight of the two components, and the temperature parameter τ in Eq.(9) is set to 0.09. Additionally, we use the same loss for the reconstruction objective, *i.e.,* $L_{cs}^R(X_{1,A}, \hat{X}_{1\to 1,A})$. Therefore, our content supervision loss is calculated on both the conversion and reconstruction patterns, and we simply add them together to obtain the final content supervision loss:

$$\mathcal{L}_{cs}(X_{1,A}, \hat{X}_{1\to 1,A}, \hat{X}_{1\to 2,A}) = \frac{1}{2}\cdot(\mathcal{L}_{cs}^T + L_{cs}^R) \quad (12)$$

We optimize the entire generator G by this loss function, forcing the model to lose less semantic content information during the VC process. To some extent, this also assists SACS-VC in decoupling the speech representations by constraining the semantic content structure.



(a) Reconstruction loss (b) Content supervision loss (c) Adversarial loss

Fig. 7. Overview of the learning strategies.

3) Adversarial loss: Following [30], the adversarial loss is adopted to synthesize realistic speech that sounds similar to the target speech. This can be written as follows:

$$\mathcal{L}_{adv}(X_{2,B},\hat{X}_{1\to2,A}) = \mathbf{E}_{\{X_{1,A},X_{2,B}\}\sim\mathcal{X}}\ \log D(X_{2,B}) + \log(1-D(\hat{X}_{1\to2,A})) \tag{13}$$

where G and D denote the generator and discriminator, respectively, and $\hat{X}_{1\to2,A} = G(X_{1,A},X_{2,B})$. The *variant loss* in WGAN-GP [50] is adopted to mitigate the training instability issue.

4) Final objectives: We train the proposed method by solving a min-max optimization problem according to the weighted sum of individual loss functions described above:

$$\min_{G}\max_{D}\mathcal{L}_{recon} + \lambda_a\mathcal{L}_{adv} + \lambda_{cs}\mathcal{L}_{cs} \tag{14}$$

where λ_a and λ_{cs} are the hyperparameters that control the relative importance of the different losses. For the experiments, we set $\lambda_a = 0.02$ and $\lambda_{cs} = 1$.

Algorithm 1: Training Strategy

Input: Multi-speaker non-parallel dataset $\mathcal{X}$, Learning rate $\eta = 0.0001$, $m = 32$, $\lambda_a = 0.02$, $\lambda_{cs} = 1$
Initialize generator $G = \{E_c, E_s, De\}$ and discriminator D,
for *number of training iterations* **do**
  for *j in 1, ..., m* **do**
    Sample source speech $X_{1,A}^{(j)} \sim \mathcal{X}$.
    Sample reference speech $X_{2,B}^{(j)} \sim \mathcal{X}$.
  Create m-sized minibatch $\{X_{1,A}, X_{2,B}\}$.
  $\hat{X}_{1\to2,A} = De(E_c(X_{1,A}), E_s(X_{2,B}))$
  $\hat{X}_{1\to1,A} = De(E_c(X_{1,A}), E_s(X_{1,A}))$
  Calculate $\mathcal{L}_{recon}(X_{1,A},\hat{X}_{1\to1,A})$,
    $\mathcal{L}_{cs}(X_{1,A},\hat{X}_{1\to1,A},\hat{X}_{1\to2,A})$,
    $\mathcal{L}_{adv}(X_{2,B},\hat{X}_{1\to2,A})$
  $\theta_D \leftarrow \theta_D + \eta\nabla_{\theta_D}\lambda_a\mathcal{L}_{adv}$
  $\theta_G \leftarrow \theta_G - \eta\nabla_{\theta_G}(\mathcal{L}_{recon} + \lambda_a\mathcal{L}_{adv} + \lambda_{cs}\mathcal{L}_{cs})$

C. Implementation details

Since our method's output is a mel-spectrogram, we implement a vocoder to achieve the transformation from the acoustic features to the speech signals. Specifically, we employed a pre-trained MelGAN vocoder [51], which is a non-autoregressive approach that performs similarly to other autoregressive vocoders. Initially, we generate the corresponding acoustic features in the required format for the MelGAN input. More precisely, we resample the audio at 22,050 HZ and perform a short-time Fourier transform (STFT) with a window size of 1024. Then, we transform the magnitude of the spectrograms into an 80-bin mel-scale and calculate its logarithm. Subsequently, these acoustic features are fed into our model to optimize its parameters. Finally, we generate the converted speech through the optimized model and vocoder.

We trained the proposed method (*i.e.*, generator and discriminator) using the ADAM optimizer (with $learning\_rate = 10^{-4}$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $weight\_decay = 10^{-4}$) for 20k iterations. The batch size is 32, and each mini-batch consists of 32 source and 32 reference utterances, which are in one-to-one correspondence. The generator and discriminator are optimized alternately in each iteration. Algorithm 1 summarizes the entire training strategy. The base code can be found on: https://github.com/XXxin1/SACS-VC.

TABLE I
NUMBER OF UTTERANCES AND SPEAKERS IN THE EXPERIMENTAL SETTING.

	Training	Validation	Testing
Speakers	99	99	10
Utterances	23595	2573	2515

TABLE II
OBJECTIVE EVALUATION RESULTS.

(a) Many-to-many setting

Methods	Similarity ↑	MCD ↓	WER ↓
MelGAN (Vocoder)	0.932	3.69	12.06
AdaIN-VC	0.749	5.97	40.68
AutoVC	0.747	6.10	23.17
VQVC+	0.766	5.91	53.39
AGAIN-VC	0.723	6.05	34.01
SACS-VC (Ours)	0.781	5.70	22.70

(b) Any-to-any setting

Methods	Similarity ↑	MCD ↓	WER ↓
MelGAN (Vocoder)	0.933	3.66	12.86
AdaIN-VC	0.752	6.12	43.63
AutoVC	0.694	6.24	26.25
VQVC+	0.735	5.98	57.75
AGAIN-VC	0.725	6.11	36.10
SACS-VC (Ours)	0.776	5.86	23.92

V. EXPERIMENTS

A. Experimental settings

The entire CSTR VCTK Corpus [25], which includes approximately 44 hours of audio from 109 different speakers and different sets of utterances, was used to train the proposed method. We randomly sampled 5 female and 5 male speakers as our unseen test speakers [1]. For each of the remaining 99 speakers, we used 90% of the utterances for training and the remainder for validation. We first trimmed the audio and transformed it into acoustic features. We randomly cropped the acoustic features with a segment window length of 128 to create batches for training. In the inference stage, VC can still easily handle variable-length inputs by virtue of our fully-convolutional architecture. The dataset details are listed in Table I.

Any-to-any VC requires that we process any speaker utterances when they are not present in the training data. Following [19], we consider two VC settings in our experiments: (1) many-to-many (**m2m**), which implements VC between speakers in the VCTK training data. These test pairs originate from the validation set described above. (2) any-to-any (**a2a**), which considers the VC between speakers that are not present in the training data. These test pairs come from the testing set described above. In both cases above, the test pairs are sampled fairly and randomly in four dimensions (intra/inter-gender). We ensure that each test pair included only 1 reference utterance. In this more challenging experimental environment, we can easily generalize the proposed method to unseen speakers without retraining or finetuning to improve the generalization ability.

Next, we compared our method against four state-of-the-art any-to-any VC methods. Indeed, We identified a comprehensive set of alternative methods and selected some of the most representative ones. These include AdaIN-VC [11], AutoVC [12], VQVC+ [13], and AGAIN-VC [18]. For a fair comparison, we reproduced their performance using the available open-source implementations with the same training data. For each method, we used the same acoustic features for training and adopted the MelGAN vocoder [51] to reconstruct the acoustic feature to waveforms.

[1]The unseen speakers are composed of female: p239, p257, p266, p295, p303 and male: p245, p251, p255, p271, p345.

B. Evaluation metrics

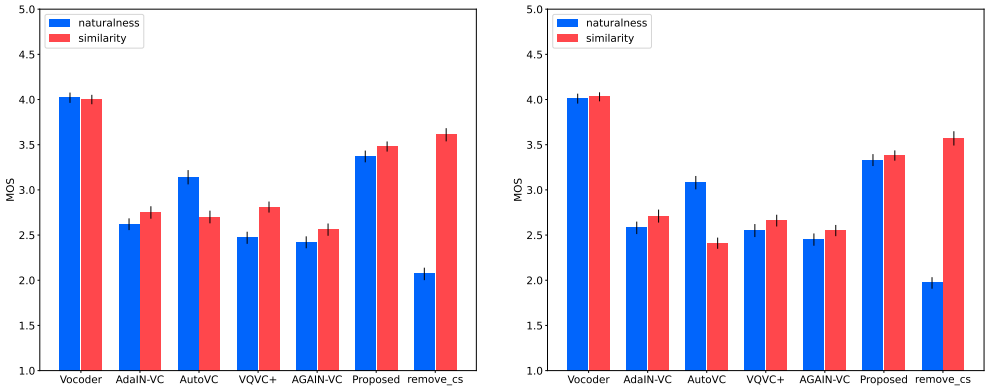
1) Subjective metrics: Following previous analyses [52], we also evaluated the naturalness of the generated speech and the similarity of the converted speech to the reference utterance (vocoder-reconstructed) in speaker timbre. The different measurements of the converted speech form our subjective metrics, *i.e.*, speech naturalness and speaker similarity. the Mean Opinion Score (MOS) was used to evaluate both perceptual qualities of the converted speech. To evaluate the speech's naturalness, the annotators in the perceptual study were asked to score the generated samples from 1 to 5 according to how natural the converted speech sounded to them. To measure speaker similarity, each annotator was presented with two audios (converted speech and corresponding reference utterance) and asked to rate them from 1 (poorest) to 5 (best) according to their confidence that the two audios originated from the same speaker. These subjective evaluations were conducted anonymously and randomly, and we ensured that there were no less than 10 annotators for each sample evaluation.

We randomly sampled 80 pairs from both the m2m and a2a sets considering all potential VC situations (intra/inter-gender) fairly. For each pair, we obtained VC using alternative methods. It is worth noting that these test pairs originated from different speakers with different transcriptions, and all methods used the same vocoder to reconstruct the audio waveforms.

2) Objective metrics: To objectively measure the quality of the generated speech, we use the similarity, Mel-Cepstral Distortion (MCD) [53], and Word Error Rate (WER) metrics. The authentic utterances are synthesized with ground-truth mel-spectrograms using MelGAN. The metrics employed are introduced below.

Similarity. The measurement of the speaker's similarity is similar to the subjective evaluation methods mentioned above. The goal is to measure whether the converted voice belongs to the target speaker of the reference utterance. For a fair and objective comparison, we employed a third-party pre-trained speaker verification system Resemblyzer[2] to embed the speaker timbre characteristics into a fixed-dimensional feature. The evaluation scores were generated by calculating the feature similarity between the speaker representations of the reference (vocoder-reconstructed) and generated utterances. The maximum similarity score is 1, and the higher the score,

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



(a) Many-to-many setting (b) Any-to-any setting

Fig. 8. MOS results on speech naturalness and speaker similarity for both many-to-many VC (left) and any-to-any VC (right), where the error bars denote a 95% confidence interval.

the more confident we are the sound came from the same speaker.

In the many-to-many and even any-to-any VC cases, 2000 testing pairs with different transcriptions and speakers were sampled from the m2m and a2a sets.

MCD. MCD measures the differences between two sequences of mel-cepstra. It requires a temporal alignment for the two input sequences. To reasonably compare the generated and ground-truth speech, we applied the Dynamic Time Warping (DTW) algorithm to align the speech audio signals [54] before calculating MCD. Here, we extracted mel-cepstrals features (MECP) from the waveform of utterances to describe the speech signals instead of the mel-spectrogram originally used. The smaller the distance, the better the conversion quality.

As the MCD calculation requires a temporal alignment between the converted and authentic reference utterances, we sampled another 2000 speech pairs from both the a2a and m2m sets, where each pair was provided with the same content but different speakers.

WER. To measure the degree the generated speech maintains the semantic content of the original during VC, we evaluate the WER of the converted utterances. This is achieved by utilizing a pre-trained automatic speech recognition (ASR) system. Here, we adopted the pre-trained WeNet [55] ASR model. Since the ASR system predicts the transcriptions, the WER can be calculated by comparing the predicted and ground-truth utterances. A lower WER value indicates that the conversion preserves more linguistic content in VC. Here, it provides evidence of the conversion quality.

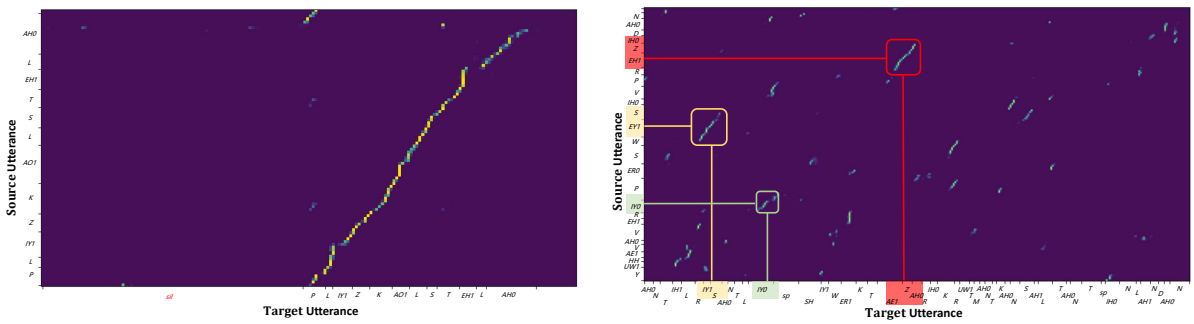
Opposing similarity, WER can measure the completeness of the semantic content, which is an important VC attribute. The 2000 conversion test pairs were sampled from the same speakers but with different linguistic content. This is a simple but effective way to measure the extent to which content is retained and the degree of disentanglement between different speech representations.

C. Experimental results

1) Subjective performance: As depicted in Fig. 8, the two MOS scores are determined with 95% confidence intervals in both the m2m and a2a settings. "Vocoder" means that the audio is synthesized from the MelGAN vocoder with the real mel-spectrogram to be considered as the upper bound of these comparative methods. The results infer that the proposed SAVS-VC performs better than the competitor baseline methods on both speech naturalness and speaker similarity, indicating better subjective conversion quality according to human perceptual evaluations. Meanwhile, the MOS results imply that our model can easily extend to conversions between unseen speakers without significant performance degradation. We also conducted related experiments without content supervision, revealing that content supervision is important to obtain more naturally converted utterances at the price of slightly degrading speaker similarity. In summary, our approach can transfer the speaker timbre well while retaining as much content information as possible. The generated audio samples are available at our demo page[3].

2) Objective performance: Table II reports the results based on the objective assessment described above. Specifically, our method achieved the best results on Similarity, MCD, and WER scores in both the m2m and a2a settings compared with the alternative any-to-any VC approaches. This is because our speaker adaptation module can automatically explore acoustically similar speech fragments, and the generated speaker representations are more compatible with the content information than alternative global speaker embeddings. Despite the slight performance degradation, SACS-VC remains more efficient than the alternative methods for the any-to-any setting. AdaIN-VC and AGAIN-VC are robust to unseen speakers regarding speaker similarity, but AutoVC and VQVC+ have significantly reduced performance when encountering unseen speakers. When disentangling the content and speaker representations to achieve VC, AdaIN-VC, AGAIN-VC, and VQVC+ lose significant amounts of content information with a higher WER score. This is because these methods lack supervision con-

[3]https://xxxin1.github.io/DEMOS-SACS-VC/



(a) Utterances with same content

(b) Utterances with different content

Fig. 9. Attention visualization results of two example pairs: (a) utterances with same content and different speakers and (b) utterances with different content and different speakers.

cerning content consistency during the conversion process. In contrast, our method can effectively preserve semantic content information due to the additional use of content supervision. Although AutoVC achieves competitive performance in WER, our method significantly outperforms it in the remaining metrics.

In conclusion, our method affords a better trade-off between speaking timbre transference and semantic content preservation. Thus, the converted utterances have better quality regarding both speech naturalness and speaker similarity.

D. Attention analysis

To present meaningful insights into the performance of the speaker adaptation module, we visualized the attention map to analyze its efficacy. However, to display an explainable visualization and focus purely on the speaker adaptation module, we eliminated the content supervision and retrained the whole method. Indeed, the final speaker adaptation module in the decoder was selected, and we sampled two example pairs from the test set considering two scenarios: a) different speakers with the same utterance and b) different utterances.

In Fig. 9 (a), for source and target utterances with the same content but spoken by different speakers, an approximately diagonal attention pattern (besides the silence part) can be seen. This is because they have a chronologically similar phonetic structure. In Fig. 9 (b), we selected a different pair with different content and different speakers. Again, the speaker adaptation module forces on the acoustically similar speech fragments (*e.g.*, /EY1 S/ and /IY1 S/ in the yellow box, /IY0/ and /IY0/ in the green box, and /EH1 Z IH0/ and /AE1 Z AH0/ in the red box). These visualization results indicate that our speaker adaptation module can explore more fine-grained voice fragments and be used to fuse more suitable speaker representations.

E. Ablation studies

We conducted ablation studies to demonstrate the effectiveness of our method's components by dropping each of them (*i.e.,* content supervision, speaker adaptation, and U-Net-like architecture design). Note that all of the presented results were generated using the same metrics and any-to-any VC setting. Table III reports the corresponding evaluation results.

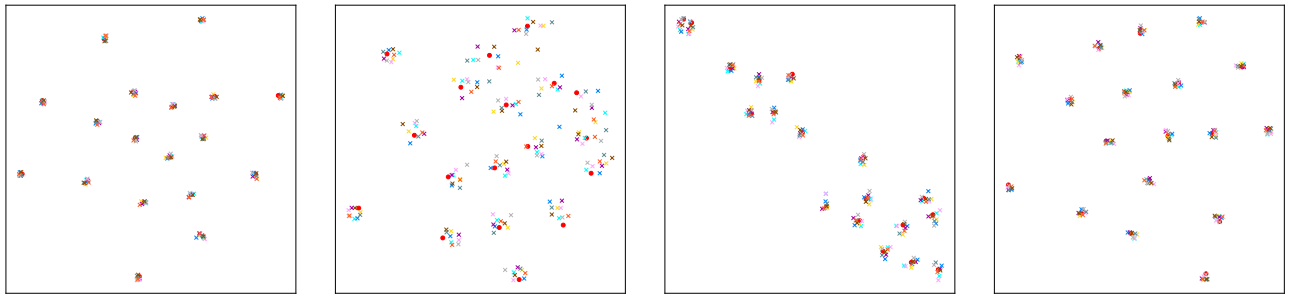
TABLE III
ABLATION RESULTS FOR THE ANY-TO-ANY VC SETTING.

Methods	Similarity ↑	MCD ↓	WER ↓
w/o $\mathcal{L}_{cs}$	0.845	5.33	91.12
- w/o $\mathcal{L}_{contrast}$	0.833	5.34	78.58
- w/o $\mathcal{L}_{content}$	0.812	5.53	57.85
w/o SA	0.728	6.09	22.78
w/o U-Net	0.836	5.39	81.10
SACS-VC (Ours)	0.776	5.86	23.92

Once we removed content supervision, the WER score significantly increased from 23.92% to 91.12%, indicating that the $\mathcal{L}_{cs}$ loss is indispensable in enforcing that the converted speech maintains the same semantic content as the source speech. Without content supervision, the content and style become unbalanced, and the model is free to excessively transform the speaker timbre without considering content consistency. This improvement in the Similarity score coincides with our intuitions. The MCD score also improves when the content supervision loss is removed. Since the MCD metric requires parallel data, all phonemes are present in the reference speech. Based on our SA module, the method can also easily achieve a diagonal attention pattern, and the related experiments are described in Sec. V-D. We argue that content supervision is essential to ensure that our method prevents over-styling and loss of semantic content. Moreover, it helps locate a better trade-off between preserving content and transferring timbre.

We also conducted ablation studies of the different objectives regarding content supervision learning. When $\mathcal{L}_{contrast}$ was removed, we only considered the value error at the corresponding feature position. We observed a slight decrease in the WER score, but the converted speech was still blurred and distorted. By removing $\mathcal{L}_{content}$, we only used contrast learning to associate the converted speech and source speech. Although the converted speech has similar pronunciations to the source speech, the experimental results in a more sophisticated testing environment (Neural-ASR) were poor. Only if we consider both learning objectives simultaneously can we maintain more semantic content information and thus achieve a lower WER score.

Additionally, we performed ablation analysis on the SA



(a) Proposed (b) w/o cs (c) w/o contrast (d) w/o content

Fig. 10. Visualization of embedding given by the content encoder. We split the final content features according to the temporal locations and visualize them. Each red point represents the content embedding from one source utterance. Each $\times$ symbol represents the content embedding of the converted speech, and different colors indicate the different utterances used as the reference for VC.

module and the U-Net [24] multi-scale architecture to demonstrate their effectiveness. To remove the speaker adaptation module, we first took the mean of the speaker features among the temporal axis, and then, we broadcasted and concatenated this averaged feature according to the length of each immediate feature in the decoder. Since this is a global and averaging speaker information modeling strategy, we trained this model as the alternative to removing the SA module. The biggest change is that both Similarity and MCD have deteriorated. Although WER is slightly increased, the more fine-grained speaker representations extracted by SA make the converted speech sound more similar to the target speaker. To remove the U-Net architecture design, we removed the intermediate features of the content encoder so that the corresponding skip connection in the decoder is canceled, allowing the degenerated content supervision to be calculated by only the final output of the content encoder. This inevitably increases the learning difficulties for preserving the semantic information, leading to a higher WER. Besides, the Similarity and MCD metrics improve because the model is more inclined to transform the target timbre than preserve the source content. Nevertheless, the too-high WER still makes the model unacceptable.

F. Visualization of content representations

To further demonstrate that our content supervision method ensures the consistency of semantic content during VC, the content representations extracted using the content encoder were visualized using t-SNE [56]. Specifically, we tested 10 unseen speakers, where we randomly sampled one utterance from different speakers, selected one sample as source speech, and used the remainder as a reference to perform VC. Ultimately, we extracted the content representations of one source speech and nine converted speech samples from the content encoder. Subsequently, we split these representations along their time axis to obtain a single concatenated embedding vector representing a speech patch. Finally, we projected all the individual concatenated embeddings into a 2-dimensional space using the t-SNE algorithm.

Fig. 10 illustrates the visualization results, where each red point represents the embedding vectors of source speech, and each $\times$ symbol indicates the representation of the converted speech. The different colors represent the converted results obtained from different reference utterances. It is clear that the content embedding vectors of the converted speech are almost completely overlapped with the corresponding source speech, as each cluster was independent of the others, with low similarity in the projected feature space. This result indicates that the content supervision method effectively preserves the consistency of semantic structure and enforces the model to create distinguishable embeddings. Moreover, it can lead the content encoder to decompose clean and accurate representations. After removing the loss $\mathcal{L}_{cs}$, the embedding vectors were found to be cluttered and overlapped in the embedding space. Specifically, the content representations of the converted speech were distant from the corresponding source speech, implying significant content distortion when $\mathcal{L}_{cs}$ is removed. This finding corresponds to the ablation results presented in Sec. V-E.

When removing only the loss $\mathcal{L}_{contrast}$, the distances within the clusters decreased compared with those obtained when $\mathcal{L}_{cs}$ was removed. Some speech clusters have closer inter-cluster distances, but this may indicate that the content encoder cannot distinguish some phonemic content. When only the loss $\mathcal{L}_{content}$ is removed, we can still maximize the mutual information between the content embedding vectors of the converted speech and source speech by using $\mathcal{L}_{contrast}$. The visualization results are very similar to those obtained with our the full version of the proposed model (without ablation of the individual losses). This implies that $\mathcal{L}_{contrast}$ plays the most important role in preserving the semantic content. However, without the error values between the corresponding embedding vectors, the intra-cluster distances increased compared with the full model. These subtle differences will lead to phoneme recognition errors, especially in the Neural-ASR system. To summarize, these visualization results demonstrated the importance and effectiveness of content supervision.

VI. Conclusions

This paper proposed a novel method for any-to-any VC, SACS-VC, which attempted to solve two major problems existing voice transfer systems suffer from. Specifically, first, we adjusted the speaker distribution according to the content distribution by considering their local similarity. Second, we preserved the consistency of the semantic content in a self-supervised manner. The developed method can generate a

high-quality voice by achieving a trade-off between semantic content preservation and speaker timbre transference. Experimental results verified that our proposed method achieved comparable or better performances than current state-of-the-art any-to-any VC approaches.

1) Strengths: In any-to-any VC, very few methods have explicitly ensured the consistency of semantic content before and after conversion. However, we rearranged the speaker distribution by considering the local similarities between the source and reference utterances. Higher audio quality can be attributed to the use of the proposed framework.

2) Weaknesses: In our method, the speaker adaptation modules need to capture the local semantic similarities between the source and reference utterances. However, noise inevitably occurs when the reference utterance is too short or its linguistic content is very far from the source utterance. This is because the reference utterance contains insufficient relevant information (phonetic elements). Such noise may degrade speaker information from fine-grained features to global information, impairing conversion performance.

3) Future Work: To further improve our method, future research should focus on obtaining more suitable speaker information and producing more perceptually satisfying results. Additionally, we will explore more highly customizable VC based on multiple facets of speech, including timbre, pitch, and rhythm. Building a VC system using language models and discrete tokens [57], [58] is also an interesting direction we wish to explore.

REFERENCES

[1] M. Abe, S. Nakamura, K. Shikano, and H. Kuwabara, "Voice conversion through vector quantization," *Journal of the Acoustical Society of Japan (E)*, vol. 11, no. 2, pp. 71–76, 1990.

[2] A. Kain and M. W. Macon, "Spectral voice conversion for text-to-speech synthesis," in *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP'98 (Cat. No. 98CH36181)*, vol. 1. IEEE, 1998, pp. 285–288.

[3] S. H. Mohammadi and A. Kain, "Voice conversion using deep neural networks with speaker-independent pre-training," in *2014 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2014, pp. 19–23.

[4] L.-H. Chen, Z.-H. Ling, L.-J. Liu, and L.-R. Dai, "Voice conversion using deep neural networks with layer-wise generative training," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 12, pp. 1859–1872, 2014.

[5] L. Sun, S. Kang, K. Li, and H. Meng, "Voice conversion using deep bidirectional long short-term memory based recurrent neural networks," in *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2015, pp. 4869–4873.

[6] T. Kaneko, H. Kameoka, K. Hiramatsu, and K. Kashino, "Sequence-to-sequence voice conversion with similarity metric learned using generative adversarial networks." in *INTERSPEECH*, vol. 2017, 2017, pp. 1283–1287.

[7] J.-c. Chou, C.-c. Yeh, H.-y. Lee, and L.-s. Lee, "Multi-target voice conversion without parallel data by adversarially learning disentangled audio representations," *arXiv preprint arXiv:1804.02812*, 2018.

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

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

[10] M. Zhang, Y. Zhou, L. Zhao, and H. Li, "Transfer learning from speech synthesis to voice conversion with non-parallel training data," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 1290–1302, 2021.

[11] J.-c. Chou, C.-c. Yeh, and H.-y. Lee, "One-shot voice conversion by separating speaker and content representations with instance normalization," *arXiv preprint arXiv:1904.05742*, 2019.

[12] K. Qian, Y. Zhang, S. Chang, X. Yang, and M. Hasegawa-Johnson, "Autovc: Zero-shot voice style transfer with only autoencoder loss," *arXiv preprint arXiv:1905.05879*, 2019.

[13] D.-Y. Wu, Y.-H. Chen, and H.-Y. Lee, "Vqvc+: One-shot voice conversion by vector quantization and u-net architecture," *arXiv preprint arXiv:2006.04154*, 2020.

[14] K. Qian, Y. Zhang, S. Chang, M. Hasegawa-Johnson, and D. Cox, "Unsupervised speech decomposition via triple information bottleneck," in *International Conference on Machine Learning*. PMLR, 2020, pp. 7836–7846.

[15] 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.

[16] D.-Y. Wu and H.-y. Lee, "One-shot voice conversion by vector quantization," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 7734–7738.

[17] 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," *arXiv preprint arXiv:2106.10132*, 2021.

[18] Y.-H. Chen, D.-Y. Wu, T.-H. Wu, and H.-y. Lee, "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, 2021, pp. 5954–5958.

[19] Y. Y. Lin, C.-M. Chien, J.-H. Lin, H.-y. Lee, and L.-s. Lee, "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, 2021, pp. 5939–5943.

[20] J.-h. Lin, Y. Y. Lin, C.-M. Chien, and H.-y. Lee, "S2vc: A framework for any-to-any voice conversion with self-supervised pretrained representations," *arXiv preprint arXiv:2104.02901*, 2021.

[21] 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 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 6552–6556.

[22] T. Li, Y. Liu, C. Hu, and H. Zhao, "Cvc: Contrastive learning for non-parallel voice conversion," *arXiv preprint arXiv:2011.00782*, 2020.

[23] T. Park, A. A. Efros, R. Zhang, and J.-Y. Zhu, "Contrastive learning for unpaired image-to-image translation," in *European Conference on Computer Vision*. Springer, 2020, pp. 319–345.

[24] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

[25] C. Veaux, J. Yamagishi, and K. Macdonald, "Cstr vctk corpus: English multi-speaker corpus for cstr voice cloning toolkit," *University of Edinburgh. The Centre for Speech Technology Research (CSTR)*, 2017.

[26] W.-C. Huang, T. Hayashi, S. Watanabe, and T. Toda, "The sequence-to-sequence baseline for the voice conversion challenge 2020: Cascading asr and tts," in *Proc. Joint Workshop for the Blizzard Challenge and Voice Conversion Challenge 2020*, 2020, pp. 160–164.

[27] W.-C. Huang, T. Hayashi, X. Li, S. Watanabe, and T. Toda, "On prosody modeling for asr+ tts based voice conversion," in *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2021, pp. 642–649.

[28] 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/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 1717–1728, 2021.

[29] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[30] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in neural information processing systems*, vol. 27, pp. 2672–2680, 2014.

[31] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *Proceedings of the 34th International Conference on Neural Information Processing Systems*, 2020, pp. 6840–6851.

[32] F. Fang, J. Yamagishi, I. Echizen, and J. Lorenzo-Trueba, "High-quality nonparallel voice conversion based on cycle-consistent adversarial network," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5279–5283.
[33] P. L. Tobing, Y.-C. Wu, T. Hayashi, K. Kobayashi, and T. Toda, "Non-parallel voice conversion with cyclic variational autoencoder," *arXiv preprint arXiv:1907.10185*, 2019.
[34] T. Kaneko, H. Kameoka, K. Tanaka, and N. Hojo, "Maskcyclegan-vc: Learning non-parallel voice conversion with filling in frames," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 5919–5923.
[35] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo, "Stargan: Unified generative adversarial networks for multi-domain image-to-image translation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8789–8797.
[36] S. Lee, B. Ko, K. Lee, I.-C. Yoo, and D. Yook, "Many-to-many voice conversion using conditional cycle-consistent adversarial networks," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6279–6283.
[37] T. Kaneko, H. Kameoka, K. Tanaka, and N. Hojo, "Cyclegan-vc2: Improved cyclegan-based non-parallel voice conversion," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 6820–6824.
[38] X. Huang and S. Belongie, "Arbitrary style transfer in real-time with adaptive instance normalization," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 1501–1510.
[39] T. Ishihara and D. Saito, "Attention-based speaker embeddings for one-shot voice conversion," *Proc. Interspeech 2020*, pp. 806–810, 2020.
[40] X. Xu, L. Shi, J. Chen, X. Chen, J. Lian, P. Lin, Z. Zhang, and E. R. Hancock, "Two-Pathway Style Embedding for Arbitrary Voice Conversion," in *Proc. Interspeech 2021*, 2021, pp. 1364–1368.
[41] Y. Lei, S. Yang, J. Cong, L. Xie, and D. Su, "Glow-wavegan 2: High-quality zero-shot text-to-speech synthesis and any-to-any voice conversion," *arXiv preprint arXiv:2207.01832*, 2022.
[42] V. Popov, I. Vovk, V. Gogoryan, T. Sadekova, M. S. Kudinov, and J. Wei, "Diffusion-based voice conversion with fast maximum likelihood sampling scheme," in *International Conference on Learning Representations*, 2022.
[43] Y. Wang, R. Skerry-Ryan, D. Stanton, Y. Wu, R. J. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio *et al.*, "Tacotron: Towards end-to-end speech synthesis," *arXiv preprint arXiv:1703.10135*, 2017.
[44] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
[45] W. Shi, J. Caballero, F. Huszár, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang, "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 1874–1883.
[46] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerrv-Ryan *et al.*, "Natural tts synthesis by conditioning wavenet on mel spectrogram predictions," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4779–4783.
[47] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Instance normalization: The missing ingredient for fast stylization," *arXiv preprint arXiv:1607.08022*, 2016.
[48] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. icml*, vol. 30, no. 1, 2013, p. 3.
[49] A. v. d. Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *arXiv preprint arXiv:1807.03748*, 2018.
[50] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of wasserstein gans," in *Advances in neural information processing systems*, 2017, pp. 5767–5777.
[51] K. Kumar, R. Kumar, T. de Boissiere, L. Gestin, W. Z. Teoh, J. Sotelo, A. de Brébisson, Y. Bengio, and A. C. Courville, "Melgan: Generative adversarial networks for conditional waveform synthesis," *Advances in Neural Information Processing Systems*, vol. 32, pp. 14 910–14 921, 2019.
[52] M. Wester, Z. Wu, and J. Yamagishi, "Analysis of the voice conversion challenge 2016 evaluation results." in *Interspeech*, 2016, pp. 1637–1641.
[53] R. Kubichek, "Mel-cepstral distance measure for objective speech quality assessment," in *Proceedings of IEEE Pacific Rim Conference on Communications Computers and Signal Processing*, vol. 1. IEEE, 1993, pp. 125–128.
[54] D. J. Berndt and J. Clifford, "Using dynamic time warping to find patterns in time series." in *KDD workshop*, vol. 10, no. 16. Seattle, WA, USA:, 1994, pp. 359–370.
[55] Z. Yao, D. Wu, X. Wang, B. Zhang, F. Yu, C. Yang, Z. Peng, X. Chen, L. Xie, and X. Lei, "Wenet: Production oriented streaming and non-streaming end-to-end speech recognition toolkit," in *Proc. Interspeech*, Brno, Czech Republic, 2021.
[56] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne." *Journal of machine learning research*, vol. 9, no. 11, 2008.
[57] Z. Borsos, R. Marinier, D. Vincent, E. Kharitonov, O. Pietquin, M. Sharifi, O. Teboul, D. Grangier, M. Tagliasacchi, and N. Zeghidour, "Audiolm: a language modeling approach to audio generation," *arXiv preprint arXiv:2209.03143*, 2022.
[58] C. Wang, S. Chen, Y. Wu, Z. Zhang, L. Zhou, S. Liu, Z. Chen, Y. Liu, H. Wang, J. Li *et al.*, "Neural codec language models are zero-shot text to speech synthesizers," *arXiv preprint arXiv:2301.02111*, 2023.

Xuexin Xu received his B.S. degree in computer science from Fujian Normal University, China, in 2020. He is currently a postgraduate student at the School of Informatics, Xiamen University. His research interests are speech synthesis and voice conversion.

Liang Shi received his Ph.D degree from the University of Science and Technology of China in 2004. He is now an associate professor at the School of Informatics, Xiamen University, China. His research interests are wide-reaching, including information security and machine learning.

Xunquan Chen received the B.S. degree with the major in Statistics from Chongqing University, China, in 2018. He received the M.E. degree from Kobe University, Japan, in 2021, where he is currently working toward the Ph.D. degree in computer science. His research interests include machine learning and speech synthesis. He is a Student Member of ASJ.

Pingyuan Lin received his B.S. degree in electrical engineering and automation from Putian University, China, in 2018. He is currently a postgraduate student at the School of Informatics, Xiamen University. His research interests are machine learning and object detection.

Jie Lian received his B.S. degree in Software Engineering from Fujian Normal University, China, in 2020. He is currently a postgraduate student at the School of Informatics, Xiamen University. His research interests are object detection and voice conversion.

Jinhui Chen received his Ph.D. degree (2016) in information science from Kobe University (Japan). From 2016 to 2020, he was an assistant professor at Kobe University. He was an associate professor at Prefectural University of Hiroshima from 2020 to 2023. He is currently an associate professor at Wakayama University. His research interests include pattern recognition and machine learning. He is a member of IEEE, ACM, and IEICE. He has published more than 40 publications in major journals and international conferences, such as IEEE Trans. Multimedia, IEEE/ACM Trans. Audio Speech Lang. Process., ACM MM, Interspeech etc.

Zhihong Zhang received his BSc degree (1st class Hons.) in computer science from the University of Ulster, UK, in 2009 and the PhD degree in computer science from the University of York, UK, in 2013. He won the K. M. Stott prize for best thesis from the University of York in 2013. He is now an associate professor at the School of Informatics, Xiamen University, China. His research interests are wide-reaching but mainly involve the areas of pattern recognition and machine learning, particularly problems involving graphs and networks.

Edwin R. Hancock holds a BSc degree in physics (1977), a PhD degree in high-energy physics (1981) and a D.Sc. degree (2008) from the University of Durham, and a doctorate Honoris Causa from the University of Alicante in 2015. He is currently Emeritus Professor in the Department of Computer Science at the University of York, and also Adjunct Professor and Principal Investigator - Beijing Advanced Innovation Center for Big Data and Brain Computing, Beihang University. He became a fellow of the International Association for Pattern Recognition in 2000, the Institute of Physics in 2007 and the IEEE in 2016. He is currently Editor-in-Chief of the journal Pattern Recognition, and was founding Editor-in-Chief of IET Computer Vision from 2006 until 2012. He has also been a member of the editorial boards of the journals IEEE Transactions on Pattern Analysis and Machine Intelligence, Pattern Recognition, Computer Vision and Image Understanding, Image and Vision Computing, and the International Journal of Complex Networks. He was Conference Chair in 1994 and Programme Chair in 2016 for the British Machine Vision Conference, Track Chair for ICPR (2004 and 2016) and Area Chair for ECCV (2006) and CVPR (2008 and 2014). In 1997 he jointly established the EMMCVPR workshop series with Marcello Pelillo. He was awarded a Royal Society Wolfson Research Merit Award in 2009, was named BMVA Distinguished Fellow in 2016 and received the IAPR Pierre Devijver Award in 2018. From 2016-2018 he was second vice president of the IAPR. He has published about 200 journal papers and 650 refereed conference papers. He has been a Governing Board Member of the IAPR since 2006, and is currently Vice President of the Association.