
Article:
Xu, Xuexin, Lin, Pingyuan, Shi, Liang et al. (5 more authors) (2023) Any-to-Any Voice Conversion with Multi-layer Speaker Adaptation and Content Supervision. IEEE Transactions On Audio Speech And Language Processing. ISSN 1558-7916

https://doi.org/10.1109/TASLP.2023.3306716

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

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

***Abstract*—Any-to-any voice conversion can perform among arbitrary speakers with even just one single reference utterance. Many related studies have demonstrated that it can be effectively implemented by speech representation disentanglement. On the one hand, most existing solutions fuse the style representations into the content features in a global manner without considering the difference of distributions between them. On the other hand, in the any-to-any scenario, there is no effective method to ensure the consistency of the linguistic content without text transcription and additional information extracted from additional modules. To alleviate the above problems, in this paper, we propose a novel any-to-any voice conversion method, which we refer to as SACS-VC. It combines two principal modules, which are a) Style Adaptation and b) Content Supervision. Specifically, we rearrange the style representations according to the content distribution by using a temporal attention mechanism, to obtain finer-grained style timbre information for each individual content feature. Meanwhile, we associate the converted outputs and the source utterances directly to supervise the consistency of semantic content in an unsupervised manner. This can be achieved using contrastive learning based on the corresponding and the non-corresponding locations of content features. Additionally, our method can implement by using a non-parallel speech corpus without any pretraining. Experimental results demonstrate that our method outperforms the current state-of-the-art any-to-any voice conversion systems in both objective and subjective evaluation settings.**

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

I. Introduction

VOICE conversion (VC) aims to convert speaker identity from a source utterance to that of a target speaker while simultaneously preserving the original linguistic content. This approach is widely used in many applications including personalized speech synthesis and human-computer interaction. Early work [1]–[5] focused on using aligned parallel data, *i.e.*, any speech pairs from source and target speakers share the same linguistic content and are aligned in the temporal dimension. However, these data were difficult to collect and time-consuming to align. The restricted corpus availability limits the performance and generalizability of speech conversion. These limitations have motivated research to explore non-parallel voice conversion approaches [6]–[8]. They have resulted in the construction of a deep neural network to approximate a mapping function from the source speaker domain to the target speaker domain. CycleGAN-VC [8] and StarGAN-VC [9] have both employed cycle-consistency to ensure the invertible mapping that results is identical with the source input. Although these methods can generate subjectively pleasing performance without the need for a parallel corpus, they have only resulted in a conversion process for a predefined multiple speakers set. When encountering arbitrary speakers which maybe unseen during training (outside the set of of speakers used in training), the above VC methods have only rather limited conversion capabilities.

To overcome such limitation, several any-to-any voice conversion methods have been explored [10]–[12]. In particular, most existing any-to-any VC approaches are based on speech representation disentanglement. This is an effective way to address the any-to-any conversion problem by decomposing the 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. 1 demonstrates this process. To separate speaker timbre information from linguistic content as far as possible, many techniques have been proposed. These include the information constrained bottleneck layer [11], [13], phoneme transcription guidance [14], vector quantization [12], [15], [16], normalization techniques [10], [17] and self-supervised speech representation [18], [19].

Nevertheless, most of these methods only embed the speaker timbre without considering its relevance to content, which is an average global style feature Such use of averaged style sacrifices the local phonemic style modeling capability, and processes all local content features using the same transformation function in voice conversion. For example, the pioneering work reported in [10] proposed a simple yet effective method, which applies the global mean together with the variance of the target speech to the source utterance in a deep feature space. Since required the statistics are calculated globally from a fixed-length speaker representation, the fine-grained style details and phoneme-wise patterns are largely discarded. Furthermore, silence segments affect the style representation because they contain almost no useful information. The same issue exists in AutoVC [11], which applied a pretrained

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

Xuexin Xu, Pingyuan Lin, Liang Shi, Jie Lian and Zhihong Zhang are with the School of Informatics, Xiamen University, Xiamen, 361005, China.

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

Jinhui Chen is with the Prefectural University of Hiroshima, Hiroshima, Japan.

Edwin R. Hancock is with the the Department of Computer Science, The University of York, York, YO10 5GH, UK.

This work is supported by the National Natural Science Foundation of China under Grant 62176227 and U2066213.

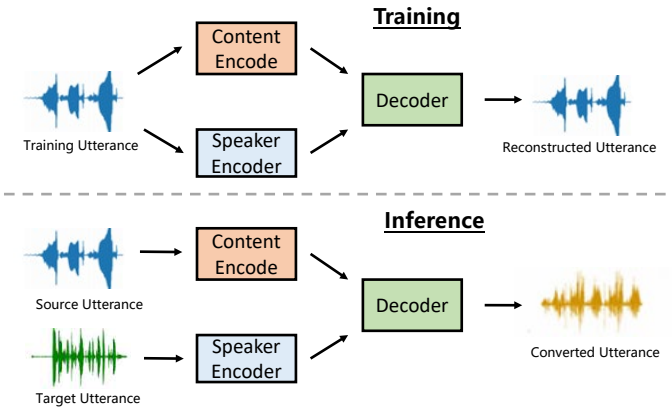


Fig. 1. An illustrative process of feature disentanglement based voice conversion.

speaker encoder to extract the global speaker timbre representations. To obtain the fine-grained style embedding for each scale of the content representation, a frequnelty used intuition is that more attention should be paid to the most similar phonemic pronunciation of the target utterances, and then extract and embed the corresponding style representation for these temporal locations.

Unfortunately, once the network probes the local fine-grained style, then unreliable style information may contaminate the corresponding content. The reason for this is that it is not possible to completely decouple the content and style. 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, and this is not acceptable in voice conversion. In fact, existing state-of-art any-to-any voice conversion, such as AdaIN-VC [10], AutoVC [11], are devoted to achieving arbitrary transfer without a parallel aligned corpus They all fail to achieve effective supervision concerning the linguistic content without any additional processing modules. The reason for this is that they only include the main objective for reconstructing input utterances (as shown in Fig. 1). Accordingly, the goal of voice conversion can be defined in a more detailed way as that of transforming the style timbre as much as possible without losing semantic content.

We attempt to address these problems and obtain a better balance between the style timbre transfer and preserving semantic content. To this end we propose a novel any-to-any voice conversion framework, which we refer to as ***SACS-VC***, which introduce ***S**tyle **A**daptation* and ***C**ontent **S**upervision* to resolve the above problems. The style adaptation module can adaptively rearrange the style timbre representations according to the content distribution using a temporal attention mechanism, and then perform style transfer on each individual content feature. We implement forthright supervision for semantic content in a self-supervised manner.

In more detail, the temporal attention map is learnt jointly from the content features and style features by implicitly aligning similar phonemic pronunciation. Subsequently, the style features are rearranged with respect to this map, and then the stylized features are generated by position-wise addition of rearranged style features to give content features. Motivated by previous research [20], for realising the content supervision, we associate the converted speech and the source input directly using contrastive learning. In other words, we maximize the mutual information of the semantic content between the converted speech and source speech. This ensures that the semantic content is preserved during the entire conversion process. Moreover, we also consider the value of the content feature error between the two. Although we only consider a non-parallel speech corpus in the training stage, we can establish the semantic correspondences between the source input and the converted output based on content features. We then maximise the correspondence in a self-supervised manner. To some extent, preserving the linguistic content can help to better decouple the speech representations. Considering the different temporal scales present in audio signals, the above operations (*i.e.*, style adaptation and content supervision) take into account different layers of the deep embedding. Meanwhile, to enhance the quality of the synthetic speech, we encapsulate the whole framework into an adversarial training strategy using a U-Net [21] like multi-scale architecture. *SACS-VC*, we can not only achieve a more fine-grained style timbre transformation for each individual phonemic content, but it can also preserve the consistency of semantic content as much as possible during voice conversion. Our main contributions can be summarized as follows:

- We propose a *style attention* module to adaptively rearrange the style distribution according to the content distribution using a temporal attention mechanism. In this way, we can generate the corresponding style features for each individual content feature. It is a more fine-grained and appropriate style pattern that depends on semantic content.
- A novel optimization objective referred to as *content supervision* is proposed. It associates converted outputs and source utterances, and helps the method to preserve the semantic content during voice conversion by maximizing the mutual information between them.
- We consider both high-level and low-level deep features at different temporal scales to obtain better convergence. Additionally, an adversarial strategy and a multi-scale architecture are also 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 alternative existing state-of-the-art any-to-any voice conversion methods on real-world VCTK datasets.

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

II. Related work

A. Direct transformation based voice conversion

To remove the requirement of a parallel corpus without any additional data or pretrained models, many researchers have developed methods based on using a feed-forward network to achieve a direct transformation from one speaker to another. Some work [8], [22]–[24] has proposed the use of non-parallel voice conversion networks, which can only achieve one-to-one conversion by training an independent network. Voice conversion among multiple speakers is a pivotal eneabling technology for a wide range of applications. Kameoka *et al.* extended an image-to-image translation method StarGAN [9] to develop StarGAN-VC [7]. Chou *et al.* [6] employed a two-stage training strategy and an adversarial speaker classifier to further remove speaker dependent information from linguistic representations. Lee *et al.* [25] overcame the drawbacks of CycleGAN-based methods [8], [26] by conditioning the network on the speaker, and the resulting method can perform many-to-many voice conversion using a single network.

However, the above voice conversion methods among multiple speakers cannot efficiently transfer those speakers not present in the training data. The disadvantage of these methods is the lack of ability to model unseen data.

B. Feature disentanglement based voice conversion

Recently, to address the limitations mentioned above, several studies based on speech representation disentanglement have attempted to decompose the speech into speaker and content representations. These methods can easily achieve any-to-any voice conversion by just replacing the speaker representation. Qian *et al.* proposed AutoVC [11], which used a pretrained speaker encoder and imposed a restriction on the length of the bottleneck layer. In their subsequent work [13] they considered different properties of speech. Zhang *et al.* [14] use the corresponding phoneme transcriptions to guide the extraction of linguistic representations. Vector Quantization (VQ) is employed in [15] and [12] to separate the speaker-independent features. AdaIN-VC [10] demonstrated that instance normalization can effectively remove speaker style information, and then applied adaptive instance normalization [27] to adjust the global statistics (*i.e.*, mean and variance). Ishihara *et al.* [28] generated content-dependent style information using an attention mechanism. In [29] the local and global style information are considered simultaneously. Self-supervised speech representations are employed in [18] and [19] for voice conversion. Wang *et al.* [16] used mutual information to measure the dependencies between speech representations.

Existing disentanglement voice conversion methods usually only consider the reconstruction objective in the training procedure. However, it is difficult to preserve the semantic content during the conversion process, especially when only using non-parallel data. On the other hand, many previous studies only embed the style representation into a predefined fixed-length vector, which is not particularly suitable for variable phonemic content. These methods fuse deep style features into the content features without considering the differences between the feature distributions. To alleviate these problems, in this paper, we explore a better trade-off between style timbre transfer and preserving semantic content. Specifically, we design a style adaptation module to rearrange the style distribution by considering the details of the content distribution. This ensures that the embedded style representation is most suited to the semantic content. To avoid the semantic content changing during the conversion stage, we propose a novel learning objective which constrains it using contrastive learning.

III. Methodology

Our proposed framework is based on the GAN [30]. Typically, a GAN is composed of a generator and a discriminator, and in our work, the generator is an encoder-decoder architecture. Similar to AdaIN-VC [10] and AutoVC [11], as illustrated in Fig. 1, the generator uses three modules to achieve any-to-any voice conversion. The training process only requires self-reconstruction from an input utterance, it 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 the speaker "1". C_A is the linguistic information relevant to content "A" captured from the content encoder $E_c(\cdot)$, S_1 indicates speaker information about identity "1" is generated by the speaker encoder $E_s(\cdot)$, and the Decoder $De(\cdot,\cdot)$ takes the content and style feature maps as inputs to synthesize the utterances $\hat{X}_{1\to1,A}$.

Confining our attention to the non-parallel setting, the speech pairs have different lengths. Therefore, given a source speech $X_{1,A} \in \mathcal{R}^{C\times T_A}$ and a reference speech $X_{2,B} \in \mathcal{R}^{C\times T_B}$, T_A and T_B denote the time length of the speech depends on the utterance. The conversion process 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 procedure, we can easily 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. It is worth noting that we adopt a multi-scale architecture in the content encoder. Thus C_A is a representation array where each item is captured by different layers. Additionally, to retain more information, the speaker encoder generates the speaker style features but without downsampling to preserve the original temporal scale, $i.e., S_2 \in \mathcal{R}^{C\times T_B}$.

The key idea is to decompose the speech into speaker and content representations. Unfortunately, the residual content information in style may lead to performance degradation. To mitigate this problem, we rearrange the disentangled style representation according to the content information. This is done using the style adaptation process described later on in Sec. III-A. The main learning objective of most any-to-any voice conversion methods is reconstructing the input utterances. There is no supervision for the converted speech without introducing additional modules. Unfortunately it is

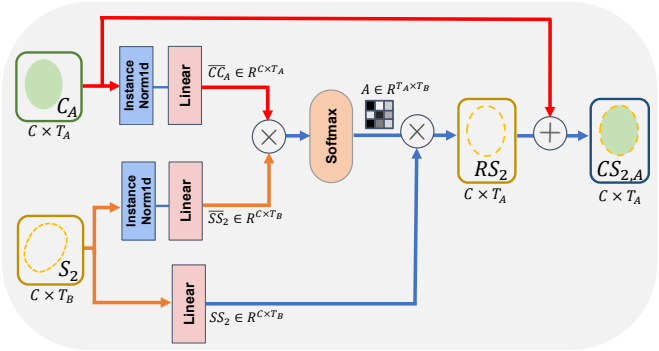


Fig. 2. Style-Adaptation module. We rearrange the style distribution according to the content information, then we fuse the content-dependent style features RS_2 into the content features C_A by point-wise addition to generate the stylized features $CS_{2,A}$.

difficult to measure the quality of the converted speech in the training stage, especially the linguistic content. To alleviate the above problem, Sec. III-B describes a content supervision approach to constrain the converted speech to be the same as the source speech. We use three types of loss function to train the entire model described in Sec. III-D, and the detailed network architecture will be discussed in Sec. III-C.

A. Style adaptation

We generate the content feature maps C_A and the style representations S_2 from source speech and reference speech through the different encoders. To overcome the negative effects of residual correlation information, the style adaptation (SA) module rearranges the style features based on their content representations, and then generates content-dependent stylized features $CS_{2,A}$.

The SA modules can automatically adapt the style distribution according to the content information. In this eraspect it is akin to an implicit alignment at the phoneme level. This adaptation can mitigate the negative effects of inconsistent content, and, moreover, it can easily achieve arbitrary voice conversion without dramatic performance degradation. The SA module is illustrated in Figure 2. Initially, given a content feature $C_A \in \mathcal{R}^{C \times T_A}$, we perform a mean-variance channel-wise normalization to remove the style information [10], and then transform it linearly to generate the normalized feature $\overline{CC}_A$. We process the style features $S_2 \in \mathcal{R}^{C \times T_2}$ in the same way to obtain the normalized style representation $\overline{SS}_2$. Meanwhile, we feed the style features S_2 into an additional linear layer, but in this case there is no normalization operation, denoted by SS_2. In a manner similar to the cross attention operation, we first calculate the correlation matrix $A \in \mathcal{R}^{T_A \times T_B}$, which can formulated as:

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

where the position (i, j) of the correlation matrix A is used to measure the relation between the i^{th} content feature and the j^{th} in style feature. In other words, for each position of the content feature, we enumerate all position of the style feature to automatically align it with the most similar phonemic position. We then rearrange the style features SS_2 by taking the product the correlation matrix A and SS_2, appropriately generate the rearranged style feature $RS_2 \in \mathcal{R}^{C \times T_A}$, we express this as follows:

$$RS_2 = SS_2 \otimes A^T \tag{4}$$

Finally, we fuse the style features into the content features to achieve voice style transfer by:

$$CS_{2,A} = RS_2 + C_A \tag{5}$$

Through the above SA process, we generate a stylized feature according to the content phonemic information, and fuse it into the content features. The generated results can automatically select an appropriate speaking style for the semantic content information that can better preserve it.

B. Content supervision

Voice conversion should fully preserve the semantic content of the source speech while transferring the speaker style. However, most existing voice conversion methods do not guarantee that such constraints are enforced without additional structures, especially in any-to-any voice conversion based on feature disentanglement. Because we cannot completely decouple the style and content from speech and ensure that they are independent of each other, incorporating style information will to some extend distort the content distribution. The resulting semantic content of the converted speech may be both distorted and ambiguous. Preserving the semantic content consistency between the converted speech and the source speech is therefore important for voice conversion. We propose the content supervision process as illustrated in in Fig. 3 to overcome this problem.

Suppose we accomplish the voice conversion taks given the source speech $X_{1,A}$ and the target speech $X_{2,B}$ coming from different speakers. Then the converted speech $X_{1\to2,A} \in \mathcal{R}^{C \times T_A}$ will be generated based on Eq. (2). The basic goal is to constrain $X_{1,A}$ and $X_{1\to2,A}$ to have the same phonemic content. Although they belong to different speakers, the semantic content should be consistent during the whole conversion process. Since we train the content encoder E_c to capture the linguistic content information of speech, the content features are readily computed from the E_c. Each layer of E_c and its location within the feature stack represents a segment or a patch of the input acoustic features (*i.e.*, speech). The deeper layers with larger receptive fields and correspond to larger patches. An intuitive idea is thereforeew to constrain the content features to be the same before and after the voice conversion at the corresponding positions. This process can be writen as :

$$\mathcal{L}_{content} = \frac{1}{L} \sum_{l=1}^{L} \|E_c(X_{1,A}) - E_c(X_{1\to2,A})\|_2 \tag{6}$$

where l denotes the l^{th} layer of the content feature stack, and we use the mean squared error (MSE) to define the content perceptual loss. This loss is very similar to the content loss in style transfer [27], but the feature extractor is not pretrained in this case. Ideally, the above approach can reduce the problem

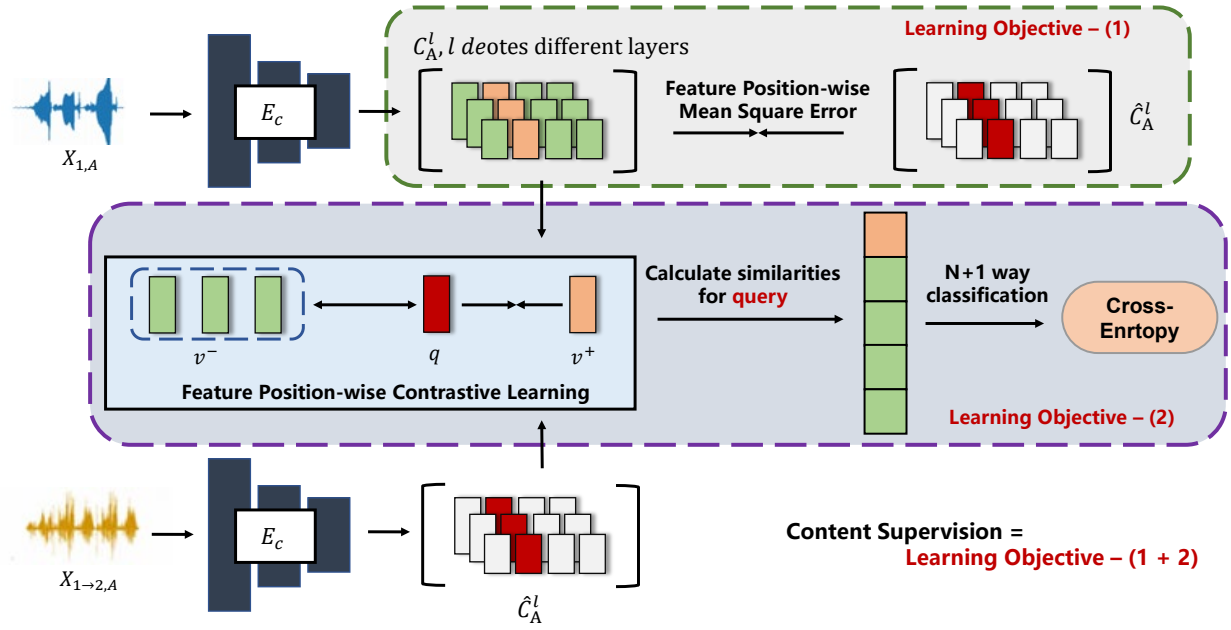


Fig. 3. The content supervision processing flow. We establish the semantic content relationships between the source speech $X_{1,A}$ and the converted speech $X_{1\to2,A}$. There are two learning objectives in our content supervision. First, we minimize the feature value errors between $X_{1,A}$ and $X_{1\to2,A}$ at the same locations. Second, we minimize the corresponding content mutual information between $X_{1,A}$ and $X_{1\to2,A}$ by using contrast learning, while encouraging the content encoder E_c to distinguish the phonemic content. The semantic content is preserved based the above objectives during the conversion process.

of content distortion or obfuscation. Unfortunately, the content encoder E_c may learn a trivial function (such as loss of ability to distinguish between phonemic content), and output the approximate representation for different semantic content. The reason for this is that we update the parameters of E_c according to the above loss without any pretraining. To avoid E_c losing the ability to capture content diversity, it is necessary to add one further requirement to make its objective multi-task.

Motivated by the unpaired image translation method based on contrastive learning in [20], we select L layers from E_c, to give a multi-layer convolution network that extracts feature stacks from the input speech spectrogram. The stack of features produced in this way can be represented as C_A^l and $\hat{C}_A^l$, where $l \in \{1, ..., L\}$, C_A and $\hat{C}_A$ denote the content features generated by $E_c(X_{1,A})$ and $E_c(X_{1\to2,A})$ respectively. Unlike the pixels in an image, the number of fragments of speech is much smaller. Thus, all temporal locations of the content features in each layer will be used.

To encourage the semantic content of converted speech to be similar to the source speech, we maximize the mutual correspondence information between them based on the InfoNCE loss [31]. Based on Eq.(6), we add a new learning objective to avoid the content encoder degrading. This objective distinguishes the different features having different temporal positions (*i.e.,*.iIt associates corresponding features to one another, while disassociating them from the remainder) by contrastive learning. The idea of contrastive learning is to construct three different types of vectors, namely a) a "query" vector q, b) a "positive" vector v^+, and c) N "negative" samples v^-. These vectors are all sampled from the content features C_A^l and $\hat{C}_A^l$, thus $v, v^+ \in \mathcal{R}^C$ and $v^- \in \mathcal{R}^{N\times C}$. For all temporal positions T, there is one positive sample and the remaining N negative samples (*i.e.*, $T = N + 1$). In our context, query refers to an output content patch, positive and negatives are the corresponding and noncorresponding input. We maximize the probability of selecting positive sample v^+ over negatives. Even conducting voice conversion, it can enforce the content encoder to output a similar embedding at the same temporal position, and generate the distinguishable representations at distinct locations. This can be also viewed as a multi-classification problem with $N + 1$ classes. Therefore, the cross-entropy loss will be calculated so as to maximize the mutual information and this is achieved in turn by maximizing the probability of matching the positive sample with the query vector. We normalize each of these vectors using the $\mathbf{L}_2$ norm. The mathematical formulation can be written as follows:

$$\ell(q, v^+, v^-) = -log\left(\frac{exp(\frac{q\cdot v^+}{\mathcal{T}})}{exp(\frac{q\cdot v^+}{\mathcal{T}}) + \sum_{n=1}^{N} exp(\frac{q\cdot v_n^-}{\mathcal{T}})}\right) \quad (7)$$

where v_n^- denotes the n^{th} negative samples and $\mathcal{T}$ is a temperature parameter used to scale the feature distances. Our goal is to associate the semantic content of the source input and the converted output. The query vector is sampled from the content features of the converted output, The positive sample and the negative samples are the corresponding and the non-corresponding source input at the different temporal locations. As a result the second objectives can be expressed as:

$$\mathcal{L}_{contrast} = \frac{1}{L} \cdot \frac{1}{N+1} \sum_{l=1}^{L} \sum_{n=1}^{N+1} \ell(q_l^n, v_l^n, v_l^{(N+1)\setminus n}) \quad (8)$$

where l denotes the index of the content feature stacks and N depends on l due to the different temporal scales.

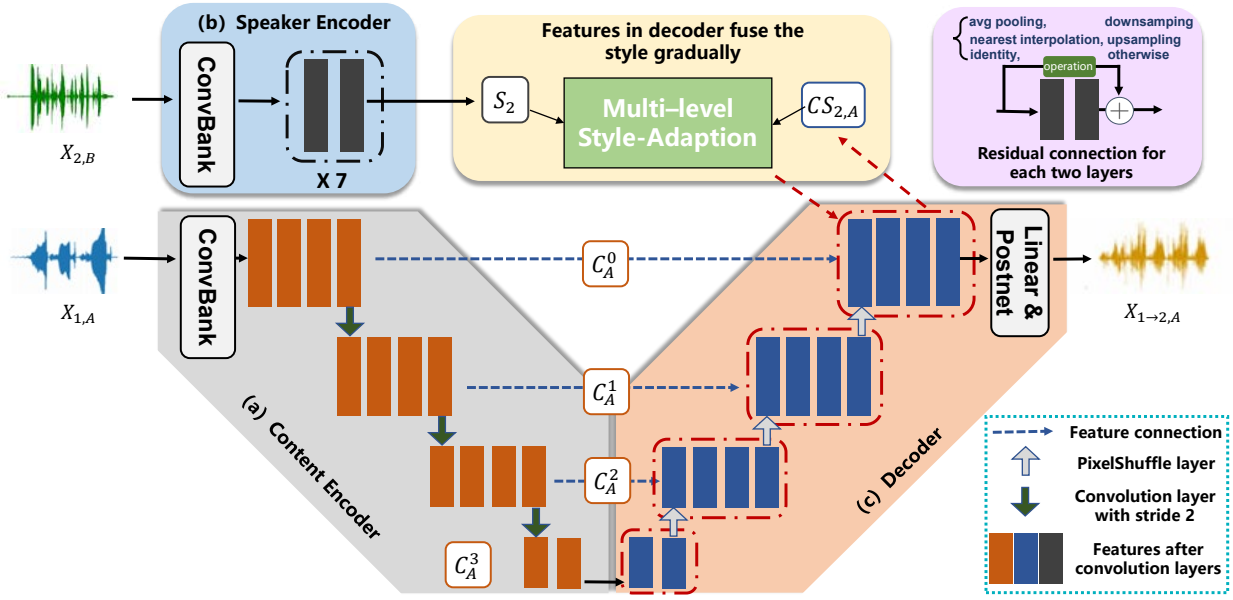


Fig. 4. An intuitive architecture diagram for the generator. (a) The content encoder. (b) The style encoder. (c) The decoder. A multi-scale architecture is implemented between (a) and (b). All the features in decoder (red dotted box) and the style features will be fed into the style adaptation modules to generate a stylized features by fusing the style information. This generating stylized features will replace the corresponding input features in decoder (red dotted arrow), and then recover the temporal scale while fusing style information gradually.

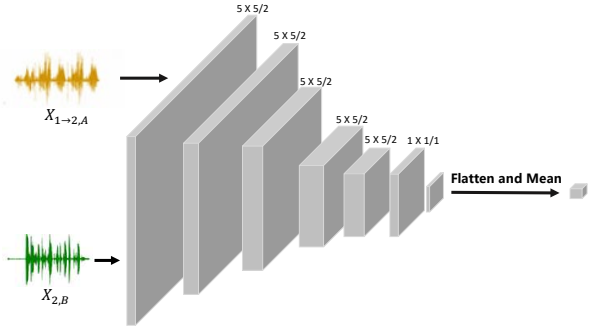


Fig. 5. An intuitive architecture diagram of the discriminator, which composes of several 2d convolution layers.

Using the above two types of constraints, we locate the mutual correspondences between the semantic content of the source speech and converted speech. We then optimize our method according to the directions of constraints. At all temporal locations, the content features of the converted output will not only be similar to the source input, but also distinguish it from alternative phonemic content. As a consequence of this content supervision, we can ensure that the semantic content information is preserved as much as possible during the entire voice conversion process.

C. Network architecture

Our framework is based on a GAN [30], which is typically composed of 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 $X_{2,B} \in \mathcal{R}^{C\times T_B}$ with different speakers. The generator G can sequentially generate the converted speech $X_{1\to2,B} = G(X_{1,A}, X_{2,B})$, which has similar content to $X_{1,A}$ and a similar timbre to $X_{2,B}$. The discriminator distinguishes a real sample of speech from a synthetic one while encouraging the generator to synthesize realistic speech of the target domain $X_{2,B}$. The network architecture is illustrated in Fig. 4 and Fig. 5.

1) The Generator: The generator G can be divided into three components, a) a content encoder E_c, b) a speaker encoder E_s, and c) a decoder D. We first obtain the high-level representations from E_c and E_s respectively, and then reconstruct the speech information through D. The generator is composed entirely of convolution neural networks to achieve non-autoregressive generation. As shown 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 the U-Net [21].

In the encoders, we first employ the ConvBank layer [32] which stacks convolution layers with different kernel sizes to enlarge the receptive field and capture long-time scale information. Subsequently, several convolution layers are applied to generate the high-level representations. The purely 1-dimensional convolution layers are implemented with a kernel size set to 5, and the stride size depends on whether downsampling of the temporal scales is required. We adapt instance normalization after each convolution layer of the content encoder to eliminate the speaking style information [10]. It is important to note that we do not downsample the temporal dimension in the speaker encoder. Instead, we keep the original temporal dimension the same as the input acoustic features to preserve the overall information. To mitigate the training difficulties, we also implement *residual connections* [33] for each pair of convolution layers with the exception of 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 scales 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.*, $C_A = \{C_A^0, C_A^1, ..., C_A^L\}$, where L denotes the numbers of downsampling operation, and the shapes of these features are $\{\mathcal{R}^{C\times T_A}, \mathcal{R}^{C\times \frac{T_A}{2}}, ..., \mathcal{R}^{C\times \frac{T_A}{2^L}}\}$. The speaker encoder embeds $X_{2,B}$ to generate the speaking style representation $S_2 \in \mathcal{R}^{C\times T_B}$, while preserving the temporal scale without any downsampling. As illustrated in Fig. 4, we set the number of downsamplings L is 3.

In the decoder, given the content features C_A and the style feature S_2, there are two main basic operations?: 1) Restoring the temporal scale from the smallest scale feature C_A^L, and 2) Fusing the style feature S_2 into the content distribution by using the style adaptation modules mentioned in III-A. A set of convolution layers with kernel size 5 and stride 1 are implemented in the decoder. For increasing the temporal resolution,a PixelShuffle1d layer [34] is used for upsampling, and local interpolation in order that the residual connections match the feature shape. A multi-scale architecture is applied to preserve increased amounts of content information. We associate the feature map after upsampling and the corresponding content representation C_A^l according to the same scale. We feed the restored feature and the style feature into the style adaptation module. Due to the speaker encoder being trained without any constraints and downsampling, we can automatically adapt and fuse the style into the converted feature gradually according to the semantic correlation. In other words, to synthesize the stylized features, a pipeline is constructed using several consecutive "1)-2)" operations to restore the temporal scale of the features and then gradually fuse the style information. Then we use a linear transformation to modify the channel to match the acoustic features. Finally, the *post network* [35] is appended but in this case without the batch normalization. This predicts a residual to add to the prediction to improve the overall reconstruction. The *post network* contains five convolution layers, where we use hyperbolic tangent activation function in all but the final layer. The channel dimension is set to 512 in the first four layers, and reduces to 80 in the final layer. We add a dropout layer with the rate set to 0.5 after each layer in the *post network*.

2) The Discriminator: Unlike the generator, the discriminator is constructed with 2D convolution layers in a manner similar to [6], [7] in order to better capture the acoustic texture. We first reshape the input speech from $\mathcal{R}^{C\times A}$ to $\mathcal{R}^{1\times C\times A}$. Subsequently, there are 5 convolution layers with stride 2 and kernel size 5×5 to gradually, downsample the feature map. The number of filters for these convolution layers are respectively 64, 128, 256, 512 and 512. To decrease the feature channel from 512 to 32, a convolution layer with unit kernel size and stride is appended. Finally, an output layer follows and is used to obtain a measure of the degree of verisimilitude of the speech in the target domain. Instance normalization [36] and Leaky ReLu activation [37] with slope 0.01 are applied after each convolution layer with the exception of the final output layer.

D. Loss function

To translate the source speech to sound like the target speaker, our proposed network is optimized through three types of loss functions in the training stage. According to Eq. (2), for given two arbitrary sampled speech instances $\{X_{1,A}, X_{2,B}\}$ from a non-parallel dataset $\mathcal{X}$, we can achieve any-to-any voice conversion based feature disentanglement.

1) Adversarial loss: Following [30], an adversarial loss is adapted to synthesize realistic speech which sounds similar to the target speech. We can write this as follows:

$$\begin{aligned}\mathcal{L}_{adv}(X_{2,B}, X_{1\to2,A}) = \mathbf{E}_{\{X_{1,A},X_{2,B}\}\sim\mathcal{X}}\ \log D(X_{2,B}) \\ +\log(1-D(X_{1\to2,A}))\end{aligned} \tag{9}$$

where G and D denote the generator and discriminator respectively, and $X_{1\to2,A} = G(X_{1,A}, X_{2,B})$. The *variant loss* in WGAN-GP [38] is adopted to mitigate the training instability issue.

2) Content supervision loss: As discussed in Sec. III-B, we use two different learning objectives to preserve the consistency of the semantic content during the speech conversion process. Thus, the *content supervision* loss depends on Eq.(6) and Eq.(8) in weighted combination *i.e.,*

$$\mathcal{L}_{cs}^{T}(X_{1,A}, X_{1\to2,A}) = \mathbf{E}_{\{X_{1,A},X_{2,B}\}\sim\mathcal{X}}\ c_1\cdot\mathcal{L}_{content}+\mathcal{L}_{contrast} \tag{10}$$

where the coefficient c_1 is set to 0.5 to determine the relative weight of the two components, and the temperature parameter $\mathcal{T}$ in Eq.(8) is set to 0.09. Additionally, we also use the same loss for the reconstruction objective, *i.e.,* $L_{cs}^{R}(X_{1,A}, X_{1\to1,A})$. Therefore, our content supervision loss is calculated on both the translation and reconstruction patterns, and we simply add them together to obtain the final content supervision loss:

$$\mathcal{L}_{cs}(X_{1,A}, X_{1\to1,A}, X_{1\to2,A}) = \frac{1}{2}\cdot(\mathcal{L}_{cs}^{T} + L_{cs}^{R}) \tag{11}$$

We optimize the entire generator G according to this loss function. This can also to some extent assist the generator to decouple the speech representations by constraining the semantic content structure.

3) Reconstruction loss: The reconstruction loss assists the generator to preserve the consistency of the spectrogram when using the same speech sample for both the input content speech and the input reference speech:

$$\mathcal{L}_{recon}(X_{1,A}, X_{1\to1,A}) = \mathbf{E}_{X_{1,A}\sim\mathcal{X}}\|X_{1\to1,A} - X_{1,A})\|_1 \tag{12}$$

where $X_{1\to1,A}$ is the self-reconstruction procedure in Eq. (1), and we use the $\mathcal{L}_1$ distance (norm) to measure the differences between the source input and the correspond reconstructed one. This reconstruction loss encourages well defined output spectrograms and ensures that the auto-encoder architecture does not loose too much information. It is also an essential part and a main objective for feature disentanglement-based any-to-any voice conversion methods [10], [11], [19].

4) Final objectives: We train the proposed method by solving a minmax optimization problem according for the weighted sum of individual loss functions described above,

$$\min_G \max_D \mathcal{L}_{recon} + \lambda_a \mathcal{L}_{adv} + \lambda_{cs} \mathcal{L}_{cs} \tag{13}$$

where λ_a and λ_{cs} are the hyperparameters which control the relative importance of the different losses. We set λ_a to 0.02 and λ_{cs} to 1 during the experiments.

E. Implementation details

The output of our proposed method is a mel-spectrogram. To this end we need to implement a vocoder to achieve the transformation from acoustic features to speech signals. We employed a pretrained MelGAN vocoder [39], which is a non-autoregressive approach but has a comparable performance with 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 the STFT (short-time Fourier transform) with STFT window size 1024. We then transform the magnitude of the spectrograms into an 80-bin mel-scale and then take its logarithm. Subsequently, these acoustic features will be fed into our model to optimize its parameters. Finally, we generate the converted speech through the optimized model and the vocoder.

We train the proposed method (*i.e.*, generator and discriminator) using the ADAM optimizer (with $learning\_rate$ = 10^{-4}, β_1 = 0.9, β_2 = 0.999, and $weight\_decay$ = 10^{-4}) for 20k iterations. The batch size is 32 and each mini-batch consists of 32 source utterances 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.

Algorithm 1: Training Strategy

Input: Multi-speaker non-parallel dataset $\mathcal{X}$,
$\eta = 0.0001$, $m = 32$, $\lambda_a = 0.02$, $\lambda_{cs} = 1$
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 a m-size minibatch $\{X_{1,A}, X_{2,B}\}$.
  $X_{1\to 2,A} = De(E_x(X_{1,A}), E_s(X_{2,B}))$
  $X_{1\to 1,A} = De(E_x(X_{1,A}), E_s(X_{1,A}))$
  Calculate $\mathcal{L}_{adv}(X_{2,B}, X_{1\to 2,A})$,
   $\mathcal{L}_{recon}(X_{1,A}, X_{1\to 1,A})$,
   $\mathcal{L}_{cs}(X_{1,A}, X_{1\to 1,A}, X_{1\to 2,A})$
  $\theta_D \leftarrow \theta_D + \eta \nabla_{\theta_D} \mathcal{L}_{adv}$
  $\theta_G \leftarrow \theta_G - \eta \nabla_{\theta_G} (\mathcal{L}_{adv} + \lambda_a \mathcal{L}_{recon} + \lambda_{cs} \mathcal{L}_{cs})$

TABLE I
NUMBER OF UTTERANCES AND SPEAKERS IN EXPERIMENTAL SETTING.

	Training	Validation	Testing
Speakers	99	99	10
Utterances	23595	2573	2515

IV. EXPERIMENTS

A. Experiments setting

The entire CSTR VCTK Corpus [40], which includes about 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 speakers and 5 male speakers as our unseen test speakers. For each of the remaining 89 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. For parallel training, we randomly cropped the acoustic features with a segment window length of 128. The details are shown in Table. I. In the inference stage, voice conversion can be easily implemented with variable-length inputs by virtue of our fully-convolutional architecture. For non-parallel voice conversion, each training pair consists of two different utterances with different content from different speakers.

Any-to-any voice conversion requires that we process any speaker utterances when they are not present in the training data. Following [18], we consider two voice conversion settings in our experiments: (1) many-to-many (**m2m**), which implements voice conversion between speakers in the VCTK training data; these test pairs came from the validation set mentioned above. Although the speakers are seen in the training stage, these utterances are not present in the training data. (2) any-to-any (**a2a**), considers the voice conversion between speakers which are not present in the training data; these test pairs came from the testing set mentioned above. In both the above cases, the test pairs were sampled fairly and randomly in four dimensions (intra/inter-gender). We ensured each test pair includes only 1 reference utterance. We can easily generalize the proposed method to unseen speakers without retraining or finetuning to improve the generalization ability.

Four comparative methods which represent state-of-the-art in any-to-any voice conversion were adopted for performance comparisons. We have identified a comprehensive set of alternative methods and selected some of the most representative ones. These include AdaIN-VC [10], AutoVC [11], VQVC+ [12], and AGAIN-VC [17]. To make fair comparison, we reproduced their performance using the available open source implementations and with the same training data. For each method, we used the same acoustic features for training, and adopted the MelGAN [39] vocoder to reconstruct the acoustic feature to waveforms.

B. Evaluation metrics

1) Subjective metrics: Following previous analyses [41], we also conducted evaluations on the naturalness of the

TABLE II
OBJECTIVE EVALUATION RESULTS.

(a) Many-to-many setting

Methods	Similarity ↑	MCD ↓	WER ↓
MelGAN (Vocoder)	0.932	3.69	15.21
AdaIN-VC	0.749	5.97	44.42
AutoVC	0.747	6.10	26.04
VQVC+	0.766	5.91	56.16
AGAIN-VC	0.723	6.05	38.10
SACS-VC (Ours)	0.781	5.70	25.91

(b) Any-to-any setting

Methods	Similarity ↑	MCD ↓	WER ↓
MelGAN (Vocoder)	0.933	3.66	15.04
AdaIN-VC	0.752	6.12	46.15
AutoVC	0.694	6.24	29.27
VQVC+	0.735	5.98	59.12
AGAIN-VC	0.725	6.11	39.51
SACS-VC (Ours)	0.776	5.86	26.46

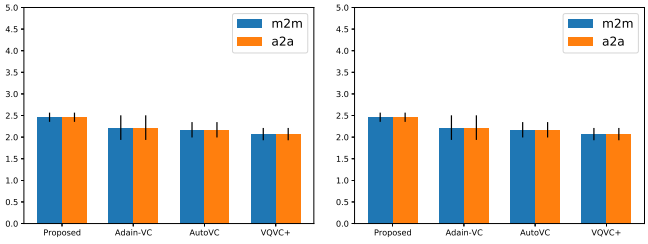


Fig. 6. MOS results on speech naturalness (left) and speaker similarity (right) for both many-to-many VC and any-to-any VC, where the bars denotes 95% confidence interval.

generated speech, and the similarity of the converted speech to the reference utterance (vocoder-reconstructed) in style timbre. The different measurements of 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 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. For measuring speaker similarity, each annotator was presented with two audios (the converted speech and the 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 set and the a2a set considering all potential speech transfer situations (intra/inter-gender) fairly. For each individual pair, we obtained voice conversion using the alternative different methods. These test pairs came from different speakers with different transcriptions. All methods studied used the same vocoder to reconstruct the audio waveforms.

2) Objective metrics: To objectively measure the quality of the generated speech, we use three different metrics, *i.e.,* a) Similarity, b) Mel-Cepstral Distortion (MCD) [42], and c) Word Error Rate (WER). The authentic utterances are synthesized with ground-truth mel-spectrograms using MelGAN. In more detail, the metrics were evaluated as follows:

Similarity. The measurement of speaker 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 pretrained speaker verification system Resemblyzer[1] to embed the speaker timbre characteristics into a fixed-dimensional feature. The evaluation scores were generated by calculating the similarity between the speaker representations of the reference utterance (vocoder-reconstructed) and the generated utterance. The maximum similarity score is 1, and the higher the score the higher the speaker confidence.

In many-to-many voice conversion, even any-to-any setting, 2000 testing pairs with different transcriptions and speakers were sampled from both m2m set and a2a set.

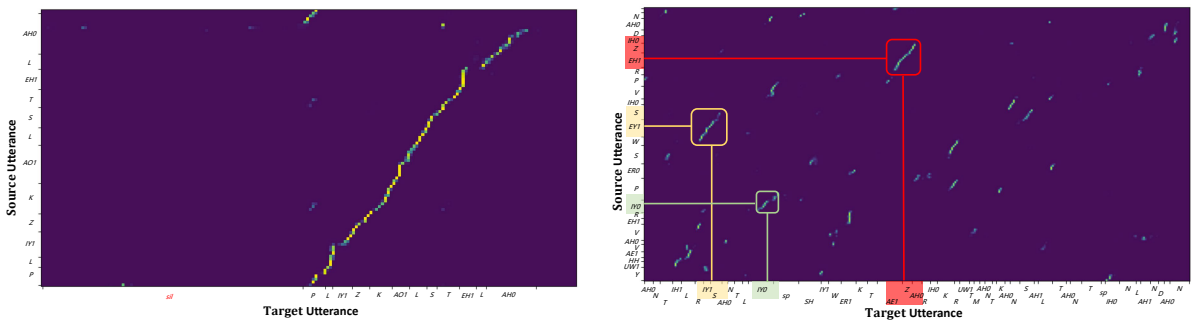
MCD. The Mel-Cepstral Distortion (MCD) is a measure the differences of two sequences of mel-cepstra. It requires a temporal alignment for the two input sequences. To make reasonable comparisons between the generated and ground-truth speech, we applied the Dynamic Time Warping (DTW) algorithm to align the speech audio signals [43] 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.

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

WER. To measure the degree to which the generated speech maintains the semantic content of the original during voice conversion, we evaluate the WER of the converted utterances. This is done by drawing support from a pretrained automatic speech recognition (ASR) system. Here, we adopted a pretrained ASR model, WeNet [44]. The ASR system can predict the transcriptions, thus the WER can be calculated by comparing the predicted and the ground-truth utterances. A lower WER value indicates that the conversion preserves more linguistic content in voice conversion. It can to some extent provide evidence of the conversion quality.

In contrast to similarity, WER can measure the completeness of the semantic content. This is an important attribute of voice conversion. The 2000 conversion test pairs are sampled from the same speakers but with different linguistic content. This is a simple but effective way to measure the extent to which

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



(a) Utterances with same content

(b) Utterances with different content

Fig. 7. Attention visualization results of two example pairs. (a) is the utterances with same content and different speakers. (b) is the utterances with different content and different speakers.

content is retained, and the degree of disentanglement between different speech representations.

C. Experimental results

1) Subjective performance: As shown in Fig. 6, the two MOS scores are determined with 95% confidence intervals in both m2m and a2a settings. Our proposed SAVS-VC performs better than other baseline methods on both speech naturalness and speaker similarity, thus indicating better subjective conversion quality according to human perceptual evaluations. Meanwhile, the MOS results imply that our model can be easily extended to conversions between unseen speakers without drastic performance degradation. Here, we also conducted related experiments for the proposed SACS-VC but without content supervision. The results indicate that content supervision is important to obtain more natural converted utterances, although at the price of slightly degrading the speaker similarity. To summarize, our approach can transfer the speaker timbre well while retaining as much content information as possible. The generated audio samples are available on our demo page[2].

2) Objective performance: Based on the objective assessment described above, the results given in Table. II were obtained. From this table when compared with the alternative any-to-any voice conversion approaches studied, our proposed method achieved the best results on Similarity, MCD, and WER scores in both the m2m and a2a settings. This may be attributed to the fact that our style adaptation module can automatically explore acoustically similar speech fragments, and the generated style representations are more compatible with the content information than alternative global style embeddings. For the any-to-any setting, despite a little performance degradation, SACS-VC remains more efficient than the alternative methods. AdaIN-VC and AGAIN-VC are robust to unseen speaker in terms of speaker similarity, but AutoVC and VQVC+ have significant reduced performance when encountering unseen speakers. When disentangling the content and style representations to achieve voice conversion, AdaIN-VC, AGAIN-VC, and VQVC+ loose significant amounts of content information with a higher WER score. This is because these methods lack supervision concerning content consistency during the conversion process. By contrast, our method can effectively preserve semantic content information due to the additional use of content supervision. Even though, AutoVC achieves competitive performance in terms of WER, our method significantly outperforms in terms of the remaining metrics.

In conclusion, our method gives a better trade-off between style timbre transference and semantic content preservation, thus the converted utterances have better quality in terms of both speech naturalness and speaker similarity.

D. Attention analysis

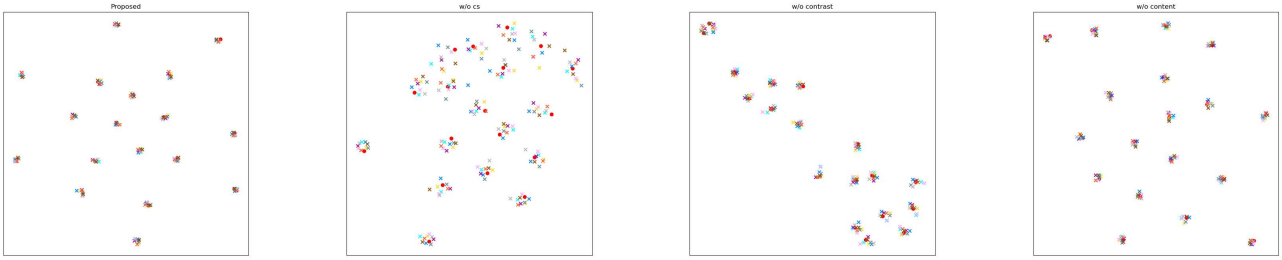
To present meaningful insights into for the performance of the style adaptation module, we visualized the attention map to analyze its efficacy. However, to display an explainable visualization and focus purely on the style adaptation module, we eliminated the content supervision loss, and retrained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

In Fig. 7 (a), the source and target utterances with the same content but spoken by different speakers show an approximately diagonal attention pattern (besides the silence part). This is because they have a chronologically similar phonetic structure. In Fig. 7 (b) we selected a different pair with different content and different speakers. Again, the style adaptation module is able to focus 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 style adaptation module can explore more fine-grained voice fragments. Moreover, it can be used to effectively fuse more suitable style representation.

E. Ablation studies

In this section, we conduct ablation studies to demonstrate the effectiveness of our proposed content supervision approach by individually dropping each of the above loss functions. This provides insights into the role of each loss function in

[2]https://www.



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

Fig. 8. Visualization of embedding given by content encoder. We split the final content features according the temporal locations and visualize them. Each red point represents the content embedding from one source utterances. Each $\times$ symbol represents the content embedding of the converted speech, the different colors indicates that the different utterances are used as reference for voice conversion.

TABLE III
ABLATION STUDIES ON ANY-TO-ANY VOICE CONVERSION SETTING.

Methods	Similarity ↑	MCD ↓	WER ↓
w/o $\mathcal{L}_{cs}$	0.845	5.33	91.80
- w/o $\mathcal{L}_{contrast}$	0.833	5.34	80.74
- w/o $\mathcal{L}_{content}$	0.812	5.53	58.46
SACS-VC (Ours)	0.776	5.86	26.46

the training stage. Note that, all of the presented results are generated using the same metrics and the same any-to-any voice conversion setting. The corresponding evaluation results are given in Table III.

Once we remove the content supervision, the WER score dramatically increased from 26.46% to 91.80%. This result indicates that the $\mathcal{L}_{cs}$ loss is indispensible in enforcing that the converted speech maintains the same semantic content as the source speech. Meanwhile the Similarity score improves. Without content supervision, the content and style become out of balance, the model is free to excessively transform the style timbre without considering the consistency of content. This improvement in 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. The method can also easily achieve a diagonal attention pattern based on our style adaptation module, and the related experiments can be found in Sec. IV-D. We argue that content supervision is essential to ensure that our method prevents over-styling and loss of semantic content. Moreover, it helps to locate a better trade-off between the content and style.

We also conducted ablation studies of the different objectives for content supervision learning. When $\mathcal{L}_{contrast}$ is removed we only considered the error at the corresponding position of the feature. We observe a slight decrease in the WER score, but the converted speech was still blurred and distorted. In removing $\mathcal{L}_{content}$, we only associated the converted speech and source speech using contrast learning. The converted speech has similar pronunciation to the source speech. However, in a more sophisticated testing environment (Neural-ASR), the results were not so good. Only if we consider both of those learning objectives simultaneously, can we maintain more semantic content information and thus achieve a lower WER score.

F. Visualization of content representations

In order to further demonstrate that our content supervision method can ensure the consistency of semantic content during voice conversion, the content representations extracted using the content encoder were visualized using t-SNE [45]. We tested 10 unseen speakers. We first randomly sampled one utterance from different speakers. We selected one sample as source speech and used the remainder as reference speech to perform voice conversion. We extracted the content representations of one source speech and nine converted speech samples from the content encoder, and then we split these representations along their time axis to obtain a single concatenated embedding vector which can be used to represent a patch in speech. Finally, we projected all of the individual concatenated embedding vectors so obtained into a 2-dimensional space using the t-SNE algorithm.

Fig. 8 presents the visualization results. 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 source speech. Each cluster indicated a certain speech patch, they were independent of each other with low similarity in projected feature space. This result indicates that the content supervision method employed can effectively preserve the consistency of semantic structure. Moreover, it can lead the content encoder to decompose representations which are both clean and accurate. After removing the loss $\mathcal{L}_{cs}$, the embedding vectors were found to be cluttered and overlapped in the embedding space. In particular, the content representations of the converted speech were distant from the corresponding source speech. This implies significant distortion of the content when $\mathcal{L}_{cs}$ is removed in the ablation study. When removing only the loss $\mathcal{L}_{contrast}$, the distances among the corresponding clusters increased compared with those obtained when $\mathcal{L}_{cs}$ was removed. However, some speech clusters are close to each other in the

embedding, and this may indicate that the content encoder looses the ability to distinguish some phonemic content. When only the loss $\mathcal{L}_{content}$ is removed, because we maximize the mutual information between the content embedding vectors of the converted speech and source speech by using this loss, the results were very similar to those obtained with our full proposed model (without ablation of the individual losses). This implies that $\mathcal{L}_{contrast}$ plays an 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 proposed 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.

V. Conclusions

In this paper, we have proposed a novel method to achieve any-to-any voice conversion, which we refer to as SACS-VC. It attempted to solve two major problems with existing voice transfer systems. Firstly, we adjust the style distribution according to the content distribution by considering the local similarity between them. Secondly, we preserve the consistency of the semantic content in a self-supervised manner. Our proposed method can generate high-quality voice by achieving a trade-off between semantic content preservation and style timbre transfer. Experiments verified that our proposed method achieved comparable or even better performances than other SOTA any-to-any voice conversion approaches.

1) Strengths: In any-to-any voice conversion, there are very few methods that have explicitly ensured the consistency of semantic content before and after conversion. We, on the other hand, rearrange the style 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 style adaptation modules needs 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 that of the source utterance. Such noise may impair the conversion performance. The reason for this is that there is insufficient relevant information contained in the reference utterance.

3) Future Work: To further improve our method, further investigation should be made into obtaining more suitable style information and producing more perceptually satisfying results. Additionally, we will explore more highly customizable voice conversion based on multiple facets of speech including timbre, pitch and rhythm.

References

[1] S. Desai, A. W. Black, B. Yegnanarayana, and K. Prahallad, "Spectral mapping using artificial neural networks for voice conversion," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 5, pp. 954–964, 2010.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] 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.

[19] 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.

[20] 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.

[21] 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.

[22] 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.

[23] 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.

[24] 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.

[25] 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.

[26] 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.

[27] 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.

[28] T. Ishihara and D. Saito, "Attention-based speaker embeddings for one-shot voice conversion," *Proc. Interspeech 2020*, pp. 806–810, 2020.

[29] 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.

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

[32] 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.

[33] 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.

[34] 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.

[35] 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.

[36] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Instance normalization: The missing ingredient for fast stylization," *arXiv preprint arXiv:1607.08022*, 2016.

[37] 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.

[38] 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.

[39] 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.

[40] 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.

[41] M. Wester, Z. Wu, and J. Yamagishi, "Analysis of the voice conversion challenge 2016 evaluation results." in *Interspeech*, 2016, pp. 1637–1641.

[42] 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.

[43] 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.

[44] 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.

[45] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne." *Journal of machine learning research*, vol. 9, no. 11, 2008.