IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 14 November 2023, accepted 7 December 2023, date of publication 12 December 2023,
date of current version 15 December 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3341434

RESEARCH ARTICLE

Perturbation AUTOVC: Voice Conversion From Perturbation and Autoencoder Loss

HWA-YOUNG PARK[1], YOUNG HAN LEE[2], AND CHANJUN CHUN[1], (Member, IEEE)
[1]Department of Computer Engineering, Chosun University, Gwangju 61452, South Korea
[2]Intelligent Image Processing Research Center, Korea Electronics Technology Institute (KETI), Seongnam 13509, South Korea

Corresponding author: Chanjun Chun (cjchun@chosun.ac.kr)

This work was supported in part by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea Government (MSIT) (Localization Technology Development on Spoken Language Synthesis and Translation of OTT Media Contents) under Grant 2022-0-00963.

ABSTRACT AUTOVC is a voice-conversion method that performs self-reconstruction using an autoencoder structure for zero-shot voice conversion. AUTOVC has the advantage of being easy and simple to learn because it only uses the autoencoder loss for learning. However, it performs voice conversion by disentangling speech information from speakers and linguistic information by adjusting the bottleneck dimension; this requires highly meticulous fine tuning of the bottleneck dimension and involves a tradeoff between speech quality and speaker similarity. To address these issues, neural analysis and synthesis (NANSY)—a fully self-supervised learning system that uses perturbations to extract speech features—is proposed. NANSY solves the problem of the adjustment of the bottleneck dimension by utilizing perturbation and exhibits high-reconstruction performance. In this study, we propose perturbation AUTOVC, a voice conversion method that utilizes the structure of AUTOVC and the perturbation of NANSY. The proposed method applies perturbations to speech signals (such as NANSY signals) to solve the problem of the voice conversion method using bottleneck dimensions. Perturbation is applied to remove the speaker-dependent information present in the speech, leaving only the linguistic information, which is then passed through a content encoder and modeled as a content embedding containing only the linguistic information. To obtain speaker information, we used x-vectors, which are extensively used in pretrained speaker recognition. The concatenated linguistic and speaker information extracted from the encoder and additional energy information is used as input to the decoder to perform self-reconstruction. Similar to AUTOVC, it is easy and simple to learn using only the autoencoder loss. For the evaluation, we measured three objective evaluation metrics: character error rate (%), cosine similarity, and short-time objective intelligibility, as well as a subjective evaluation metric: mean opinion score. The experimental results demonstrate that our proposed method outperforms other voice conversion techniques and demonstrated robust performance in zero-shot conversion.

INDEX TERMS Autoencoder, information perturbation, speech signal processing, voice conversion.

I. INTRODUCTION

Voice conversion refers to the technique used to convert a given speech into the speech of another speaker [1], [2]. The main goal of voice conversion is to replace nonverbal information, such as speaker characteristics with the desired speaker information while preserving the linguistic information in the speech signal. The converted speech follows the characteristics of the target speaker, such as gender and age. Voice conversion technology can be used for privacy purposes or in applications, such as voice animation, voice assistants, and speech synthesis [3], [4], [5].

The classical voice conversion methods include statistical methods using Gaussian mixture models (GMMs), deep learning-based methods using recurrent neural networks, convolutional neural networks (CNNs), and exemplar-based methods, such as non-negative matrix factorization [6], [7], [8], [9], [10], [11]. However, these methods typically require

The associate editor coordinating the review of this manuscript and approving it for publication was Lin Wang.

data comprising identical linguistic information from the source and target speakers' speeches for training. In practice, collecting large amounts of parallel data is difficult and time-consuming. Therefore, for voice conversion methods that use parallel data, methods that can be trained using small amounts of data have been studied. These methods use classical voice conversion methods based on GMM or deep learning in combination with various approaches, even when the amount of data is limited [12], [13], [14]. However, learning methods using parallel data require an additional procedure to perform temporal alignment between the source and target speaker's speech data. If temporal alignment is not properly performed, the quality of the converted speech will be reduced. Therefore, other voice conversion methods that do not require parallel data have been investigated.

To address the difficulty of parallel data collection, a method utilizing WaveNet and a nonparallel voice conversion method utilizing data augmentation techniques have been proposed [15], [16]. The method using WaveNet is a voice conversion method based on the vocoder's ability to reconstruct input features into waveforms. The voice conversion method using a vocoder is also presented in [17]. Voice conversion methods using data augmentation generate parallel data with acoustic features, such as duration, prosody, and energy, similar to the original voice, and then perform parallel voice conversion. Voice conversion methods that utilize nonparallel data based on deep learning have also been studied [18], [19], [20], [21], [22].

Other voice conversion methods include text-based approaches. The best-known text-based approach uses automatic speech recognition (ASR) models to extract phonetic posteriograms (PPGs), which are then used as linguistic information [23], [24]. Text-based approaches that use ASR models have accurate linguistic information, are unlikely to be corrupted during voice conversion, and can even perform voice conversion between speakers of different languages if the ASR model used supports multiple languages. For example, DeepConversion utilizes an ASR model to perform voice conversion by mapping PPGs, speaker-dependent features, and Mel-Cepstral coefficients (MCEP) [25]. However, because a large amount of parallel data is required to train an ASR model to extract PPGs used in voice conversion, there may be inevitable errors in the process of extracting PPGs (owing to insufficient data) for training in a low-resource, multilingual environment. This leads to mispronunciations of the converted speech, thus reducing the quality of the converted speech. Therefore, ASR model-based voice conversion has performance limitations in low-resource multilingual environments.

Voice conversion using the style transfer model has been studied as a method that does not require additional modules, such as ASR models and parallel data [26], [31]. The style transfer model, which was originally proposed to perform style transfer between nonparallel images, separates the morphological information of the image from the style information. By importing it into the speech domain, similar processing can be applied as in style transfer. For instance, in style transfer, the morphological information of an image is preserved, while other features (i.e., texture, color) are transformed into desired information. Similarly, in voice conversion, the linguistic information of speech is preserved, while other features present in the speech (i.e., prosody, expressiveness, formants) can be transformed into desired information. CycleGAN-VC and StarGAN-VC are examples of transfer-based voice conversion methods. CycleGAN-VC is a generative model that performs voice conversion based on CycleGAN, which aims at one-to-one mapping to convert speech between two different speakers. Unlike general generative models, it learns two-way mappings [26], [27]. As the features of the two speakers used for training were identified, natural voice conversion was possible with the voices of the speakers present in the model. However, because the features of speakers who are not used in training cannot be accurately captured, the target speaker(s) for which voice conversion can be performed are limited only to the observed speaker. This means that there is very little controllability of speaker information. To overcome the limitation of CycleGAN-VC, which allows only one-to-one mappings, StarGAN-VC (based on StarGAN) was proposed, which allows many-to-many mappings [31], [32]. StarGAN-VC can perform many-to-many voice conversions with one generator by receiving additional input from the target speaker's code consisting of a one-hot vector. However, like CycleGAN-VC, StarGAN-VC also has the limitation of being able to convert only to the speakers used in training. This means that there is still a lack of control over speaker information.

Another voice conversion method is the information bottleneck approach. The bottleneck approach is a voice conversion method that does not require parallel data or additional modules. This method disentangles the speaker information and linguistic information present in the speech features by adjusting the time or channel dimension of the speech features and then reconstructs the separated information to perform voice conversion. Unlike the style-transfer-based voice conversion method, which identifies and converts the speaker's features, it performed zero-shot voice conversion using a bottleneck to disentangle linguistic and speaker information from the source voice. However, this approach is limited in that the highly sensitive bottleneck dimension must be set heuristically, and if the incorrect bottleneck dimension is set, the quality of speech is either reduced or voice conversion is not performed well.

Finally, neural analysis and synthesis (NANSY) using perturbation was proposed to avoid the aforementioned problems [34]. Perturbation distorts other speaker features in speech information while keeping linguistic information intact; in this way, only the desired features can be controlled, thus eliminating the need for a bottleneck structure. Therefore, NANSY has a high-reconstruction performance with no trade-off between the quality of the reconstructed speech and the speaker similarity of the converted speech. To achieve fully self-supervised learning, NANSY trains an

unsupervised speaker recognition network with the first layer of wav2vec 2.0 as input [35].

Speaker recognition is a technology that identifies a speaker from a given speech signal; it constitutes a research field that promotes the extraction of unique speaker information contained in the speech signal [36]. Deep-learning-based speaker recognition studies are actively being conducted, and speaker recognition studies utilizing large-scale public data, such as VoxCeleb and VoxCeleb2, which have speech information from more than 7000 speakers, have shown high-speaker recognition performance [37], [38]. Therefore, speaker recognition networks trained with large datasets and supervised learning methods are used for the purpose of obtaining speaker information from common voice conversion methods because they ensure sufficient performance even for unseen speakers [39], [40].

In this study, we propose a method that can perform natural voice conversion, even for unseen speakers. The proposed method retains the structure of AUTOVC's autoencoder model but is inspired by NANSY's perturbation concept. Specifically, we used perturbation to disentangle the speaker and linguistic information contained in speech, instead of the bottleneck approach, which requires sensitive tuning. Perturbation removes all speaker-dependent information from speech, leaving only linguistic information [41], [42]. This ensures that the information extracted by the network does not contain unwanted information; therefore, only the information required for learning is obtained. This increases the controllability of the model by making it possible to deal directly with the information required for learning. The perturbed speech signal was modeled and used for content embedding by the model's encoder. To obtain speaker information, the proposed method uses x-vectors extracted from a pretrained speaker recognition network. The structure of the model utilized in this study performs self-reconstruction using an autoencoder structure, which enables easy and simple training, using only autoencoder loss. Furthermore, unlike NANSY, which leverages the fact that each layer of wav2vec 2.0 contains different representations, our method demonstrates the ability to separate linguistic and speaker information using only perturbation. Finally, while other zero-shot conversion methods focus on transforming to excluded speakers from the training data, this paper showcases the possibility of zero-shot conversion to speakers from entirely unrelated datasets. The process of reconstructing the converted speech into a waveform was utilized by the HiFi-GAN.[1]

II. RELATED WORKS

A. STYLE TRANSFER-BASED VOICE RECOGNITION

CycleGAN-VC and StarGAN-VC are representative models that perform voice conversion using the style transfer method [27], [28], [29], [30], [32], [33]. CycleGAN-VC is a voice conversion model based on the CycleGAN model, which originally demonstrated good performance in the field of nonparallel image-to-image translation and uses a cycle-consistent adversarial network and identity-mapping loss. Specifically, it uses cycle-consistent and adversarial losses to learn bidirectional mappings differently from typical generative models, and identity-mapping loss to prevent inputs from being output unchanged. The CycleGAN-VC generator uses a one-dimensional (1D) CNN to capture a broad range of temporal structures while preserving the structure of the input signal. CycleGAN-VC is a method that does not rely on additional modules and parallel data that exhibited better performance in the Voice Conversion Challenge 2016 (VCC 2016) than those used by a GMM-based method trained using twice as many parallel datasets [43].

However, there was a difference between the voice converted by CycleGAN-VC and the actual voice of the target speaker. CycleGAN-VC2 can reduce this difference by utilizing the 2-1-2D CNN structure. 2-1-2D CNN structure generator that employs a 2D CNN for downsampling and upsampling to minimize structural loss and a 1D CNN for the main conversion process. Finally, it used a two-step loss function, which calculated the adversarial loss for the forward and inverse cycles.

CycleGAN-VC and CycleGAN-VC2 used the Mel-Cepstrum as the input features because the direct use of the Mel-Spectrogram damages the time–frequency structure of the input. The third proposed CycleGAN-VC3 solves this problem and uses Mel-Spectrogram by introducing time–frequency adaptive normalization (TFAN). The TFAN allows the scale and bias of the transformed features to be adjusted while reflecting the source information in a time and frequency-wise manner. These calculations allow the TFAN to adapt to the time–frequency characteristics of the input signal and help preserve the time–frequency structure during voice conversion.

MaskCycleGAN-VC uses the filling-in-frame (FIF) auxiliary method, which is a complementary-based, self-supervised method that has been used in the fields of inpainting in computer vision and infilling in natural language processing [44], [45]. Parts of the Mel-Spectrogram were intentionally corrupted and restored to allow the model to grasp the time–frequency structure, thus reducing the damage to the time–frequency structure of the input signal during the conversion process.

Furthermore, a voice conversion method that applies a transformer to CycleGAN-based models to enhance their temporal dependencies was proposed [46]. This method replaces some of the residual convolution layers of the CycleGAN-VC model with a transformer layer so that the temporal dependence can be secured without deepening the model.

CycleGAN-VC is capable of natural voice conversion, but it has a limitation in that it can only convert one-to-one mapping speech. To solve this problem, StarGAN-VC was proposed, which performs many-to-many mappings using a

[1]Our code and trained models are available at https://github.com/cjchun3616/perturbation_autovc

single model. StarGAN-VC uses a single generator to receive an arbitrary target characteristic code in the form of a one-hot vector as a conditional input and converts it into the voice of the corresponding target. StarGAN-VC adds a speaker classifier to ensure that the converted voice matches the target speaker's voice. The losses used in StarGAN-VC consist of adversarial, classification, and cycle consistency losses.

However, because StarGAN-VC is trained to generate data far from the decision boundary that separates real speech from converted speech, it cannot follow the full distribution of the actual data. To alleviate this problem, StarGAN-VC2 was proposed. StarGAN-VC2 uses a source-and-target conditional generator and discriminator that provide both source and target speaker codes simultaneously. To ensure high-quality voice conversion, StarGAN-VC2 uses conditional instance normalization (CIN) to compute the scale and bias for domain feature. This can directly modulate features depending on the domain, as opposed to the conventional method. Finally, StarGAN-VC2 uses a Mel-Spectrogram as the input feature.

B. AUTOVC

AUTOVC enables zero-shot voice conversion by performing self-reconstruction using an autoencoder structure [40]. The essence of AUTOVC is to disentangle speaker information from the linguistic information in the input data by utilizing a carefully designed bottleneck dimension in the encoder. The input of the encoder is the Mel-Spectrogram of the source speaker, and speaker information of the source speaker is used as an additional input, which helps the encoder separate speaker information from the linguistic information in the input data. This separated latent vector containing only linguistic information is input to the decoder along with the speaker information of the target speaker, and the decoder learns to restore the converted speech. AUTOVC has the advantage of being very easy to learn because it utilizes only the reconstruction loss through the autoencoder structure. However, the overall performance of the model is highly dependent on the tuning of the bottleneck dimensions. For example, if the bottleneck dimension is too narrow, the latent vector exiting the encoder will likely lose some linguistic information, and the quality of speech restored by the decoder will be poor. Conversely, if the bottleneck's dimensional range is too wide, the latent vector will likely contain linguistic information and the speaker information of the source speaker; in this case, the quality of the restored speech is good, but the voice conversion is poor, and the similarity to the target speaker may be low.

C. NANSY

NANSY uses a perturbation method to synthesize improved quality speech while solving the tradeoff problem of the bottleneck approach [34]. The perturbation method randomly perturbs other speaker information while retaining the linguistic information in the original speech signal to extract only the desired information; thus, there is only one piece of information contained by each input feature, thus controlling all the information needed for synthesis. Unlike previous methods, which only considered speaker and linguistic information, NANSY considers speaker, linguistic, pitch, and energy information. The pitch and linguistic information were extracted using the newly proposed extraction method. For the pitch information, the Yin algorithm was applied to the input data with perturbation so that the pitch or information can be tracked well, even in speech with jitter or subharmonics. Similarly, the middle layer of wav2vec 2.0, with perturbed data as input, was used as the linguistic information [47]. To obtain speaker information, NANSY leverages the fact that each layer of wav2vec 2.0 has distinct representations. In NANSY's paper, the visual observation of the first layer of wav2vec 2.0 using t-SNE revealed that the first layer clusters based on speakers [48]. This observation indicates that the first layer of wav2vec 2.0 contains speaker information [49]. To leverage this insight, unperturbed speech is used as input to wav2vec 2.0, and the first layer is extracted. The extracted features are used as inputs for the speaker recognition network, which learns in an unsupervised manner for fully self-supervised learning. The x-vector from the trained speaker recognition network is used as speaker information for speech synthesis. In the speech synthesis stage, we train the source synthesis and filter synthesis parts that have the same network structures based on source-filter theory. This separation has the advantage of not only making the model more interpretable but also enabling pitch shifts while preserving formant information.

III. PERTURBATION AUTOVC

Fig. 1 shows the overall structure of the Perturbation AUTOVC used in this study. The neural network structure is composed of a content encoder, speaker encoder, and decoder. The content encoder and decoder (represented by dashed boxes) are trainable networks. The speaker encoder (represented by a solid box) is a speaker recognition network pretrained using the VoxCeleb2 dataset. The content encoder and decoder, represented by the dashed boxes, are composed of the same network structure as in the existing AUTOVC. In AUTOVC, speaker information is provided as an input to the content encoder in the form of additional information to disentangle the speaker and linguistic information contained in the speech signal; however, in this study, speaker information is not provided because a disentangling method using perturbation is used. In AUTOVC, the downsampling factor was set to 32 to create a bottleneck; however, we set it to one to avoid bottlenecks and learn bypasses. The 256-dimensional content embedding extracted from the content encoder was used as the linguistic information. For speaker information, we used 192-dimensional x-vectors (rather than conventional d-vectors) extracted using the emphasized channel attention and propagation and aggregation (ECAPA)-time delay neural network (TDNN) module, which is a speaker recognition network. In addition, inspired by NANSY's use of energy information, we concatenated

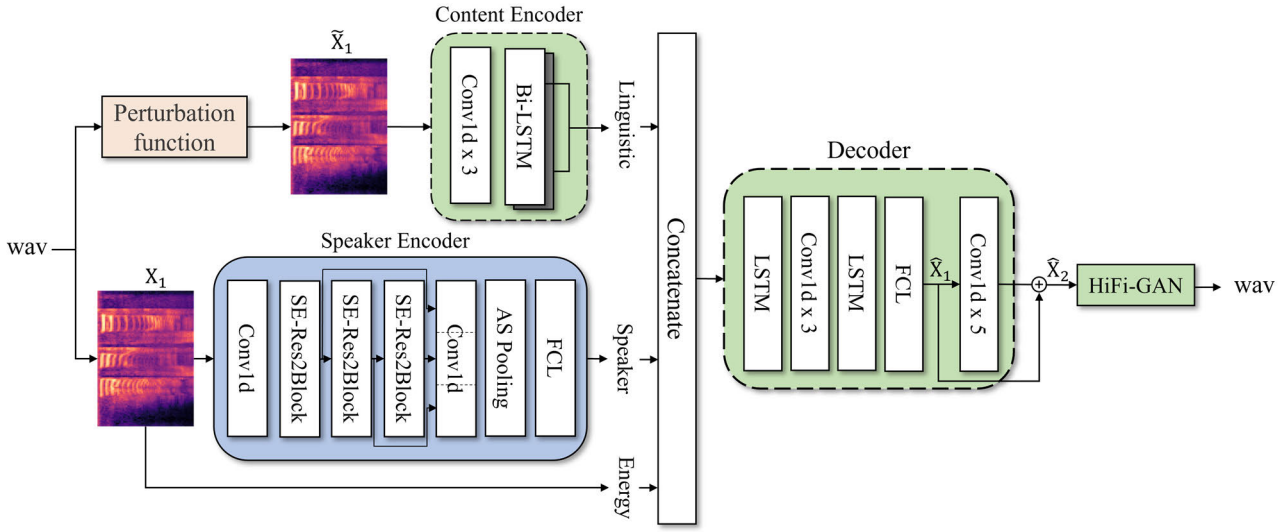

FIGURE 1. Perturbation AUTOVC architecture. The inputs to the content encoder include $\tilde{X}_1$, which is the speech with the formant, pitch, and frequency response of the perturbed speech signal, and X_1, which is the original speech signal without perturbation used to extract speaker and energy information. The term Bi-LSTM in the content encoder refers to the bi-directional long short-term memory (LSTM), and the white and gray boxes represent the forward and backward directions, respectively. The term FCL in the decoder refers to the fully connected layer. The speaker encoder has the same network structure as the recognition network ECAPA-TDNN. SE-Res2Block in speaker encoder is a network module that combines the SE Block and ResNet2 Block, and AS Pooling refers to attentive state pooling. The output of the decoder $\hat{X}_1$ is the first converted speech, and $\hat{X}_2$ is the final result after $\hat{X}_1$ passes through additional convolution layers and after the addition of the Mel-Spectrogram of the resulting speech to $\hat{X}_1$.

the linguistic, speaker, and energy information extracted from each encoder and used them as inputs to the decoder. The speech used to extract speaker and energy information was the original speech signal without perturbation. The HiFi-GAN was used as a vocoder to restore the Mel-Spectrogram to a waveform. Detailed descriptions of each submodule are provided below. Unless otherwise noted, batch normalization is applied after the convolutional layers of the submodules, followed by the activation function ReLU.

A. CONTENT ENCODER

The purpose of the content encoder is to extract linguistic information from the speech signal. In this process, we applied perturbation to the input data to remove speaker-dependent information independent of the bottleneck dimension. Based on NANSY, we applied three perturbation functions to the input signal: 1) formant shifting (*fs*), 2) pitch randomization (*pr*), and 3) random frequency shaping using a parametric equalizer (*peq*). Formant shifting is the process of adjusting the current formant frequencies by multiplying them by a shifting factor. This adjustment is achieved by manipulating the sampling frequency. For instance, to multiply all formants by a factor of 1.10 (i.e., raising them by 10 percent), a sampling frequency increases by a factor of 1.10 (without changing the samples). Afterward, the duration is lengthened by a factor of 1.10, and the pitch is lowered by a factor of 1.10 to restore the original duration and pitch. Finally, the audio can be resampled to the original sampling frequency to perform formant shifting. Shifting factor is sampled uniformly from U (1.2, 1.5) . The pitch randomization can be expressed by the following (1).

$$\begin{aligned} M_{new} &= M_{old} * \beta, \\ P_{new} &= P_{old} * \frac{M_{new}}{M_{old}}, \\ pr &= M_{new} + (P_{new} - M_{new}) * \gamma, \end{aligned} \tag{1}$$

where M_{old} represents the pitch median, P_{old} represents the pitch. β and γ represent the shifting factor and scale factor, respectively, and these are sampled uniformly from U (1.2, 1.5) ., U (1.1, 1.5). For more information about formant shifting and pitch randomization, please consult Praat [50]. The parametric equalizer is a function whose purpose is to randomly transform the frequency shaping, and can be expressed as follows,

$$H^{PEQ} = H^{LS} H^{HS} \prod_{i=1}^{8} H_i^{Peak}, \tag{2}$$

where H^{LS} represents the low-shelving, H^{HS} represents the high-shelving, and H^{Peak} is the peaking filter. By applying the three perturbation functions *fs*, *pr*, and *peq* to the original speech signal x, we can obtain $\tilde{x}$ from which speaker-dependent information is removed and only linguistic information is present.

$$\tilde{x} = fs\,(pr\,(peq\,(x)))\,. \tag{3}$$

The perturbed speech, $\tilde{x}$ is converted into an 80-dimensional Mel-Spectrogram and used as the input. It is also interesting to know whether other information distorted by perturbations affects the learning process. This can be

explained by using a self-reconstruction learning method. If we compare the self-reconstructed speech using a perturbed Mel-Spectrogram with the original speech without perturbation, the only part that will be the same is the undistorted linguistic information. Therefore, the content encoder naturally learns that distorted information is unnecessary for speech synthesis and that the content encoder learns that the content embedding output contains only linguistic information. In addition, unlike other methods whose performance is sensitive to the bottleneck dimension, the Perturbation AUTOVC is insensitive to the output dimension of the content encoder; therefore, there is no need for elaborate designs and experimentation in the effort to identify the correct bottleneck dimension. Fig. 2 demonstrates the successful separation of speaker information in the content embeddings output through the content encoder. Content embeddings extracted from the under-trained model exhibit clustering among speakers, indicating the presence of speaker information in the content embedding. In contrast, the TSNE results extracted from the well-trained model show that embeddings are evenly distributed regardless of the speaker, suggesting that the speaker information has been removed. This confirms that effective disentanglement of linguistic and speaker information within speech can be achieved without the need for fine-tuning bottleneck dimensions.

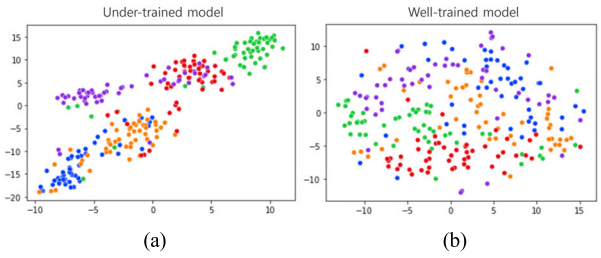

FIGURE 2. Comparison of visualization of content embedding using TSNE. The figure compares the results of content embeddings extracted from the under-trained model (a) and the well-trained model (b). The under-trained model was trained for approximately 100 iterations, while the well-trained model underwent training for about 60,000 iterations.

The perturbed Mel-Spectrogram was passed through three 5×1 convolution layers in the content encoder. The output extracted 256-dimensional content embeddings from the forward and backward sides of the Bi-LSTM, and then combined the two to use 512-dimensional content embeddings. As mentioned previously, the output dimensionality of the content encoder can be flexibly changed because it does not significantly affect the overall model performance. The content encoder of AUTOVC additionally provides the speaker information of the source speaker along with the input data so that speaker and linguistic information can be well separated, but Perturbation AUTOVC does not need to provide additional information; thus, only the perturbed Mel-Spectrogram is used as input. In addition, the Perturbation AUTOVC did not undergo the downsampling or upsampling process of content embedding by setting the downsampling factor to one. As a result, for ideal voice conversion, the content embedding in the converted speech and the content embedding extracted from the original data should be the same; therefore, the loss function of the content encoder can be defined using (4) as

$$L_{content} = \mathbb{E}\left[\left\| C_1 - E_c\left(\hat{X}_{1\to 1}\right)\right\|_1\right], \tag{4}$$

where C_1 represents the content embedding of the original data, E_c represents the content encoder, and $\hat{X}_{1\to 1}$ represents the speech restored through the decoder.

B. SPEAKER ENCODER

To obtain speaker information, NANSY utilizes the fact that the first layer of wav2vec 2.0 contains the speaker information. In addition, to extract speaker information more effectively, NANSY adopted a fully self-supervised manner to train a speaker recognition network, using the first layer of wav2vec 2.0 as input. In the field of voice conversion, the most common approach to extract speaker information is by utilizing a speaker recognition network pretrained in a supervised manner for an unseen speaker. Furthermore, there are public datasets that provide large-scale speaker data, such as VoxCeleb and VoxCeleb2, and there are many speaker recognition studies that extract d-vectors and x-vectors using these datasets. Speaker recognition networks trained in this way yield high levels of performance even when used for unseen speakers [51], [52]. Therefore, in this study, we trained a speaker recognition network using supervised learning and then extracted speaker information into a network with good embedding performance. The speaker recognition network used was the ECAPA-TDNN model, which uses a TDNN network to extract the x-vectors [53]. This model is characterized by the ECAPA, which emphasizes the interaction of input features and can perform feature extraction that is consistent across the entire network while maintaining temporal information. In addition, based on the SE-res2block structure, the output of the previous layer can be used as a skip connection structure to utilize multilayer information. As mentioned in [54], considering that shallower feature maps can yield more robust speaker embeddings, we extracted and used a 192-dimensional x-vector as our speaker embedding, a departure from the commonly employed 256- or 512-dimensional embeddings. The Voxceleb2 dataset was used for training, and ADAMW was used as the optimizer. Fig. 3 shows the embedding results for the 10 unseen speakers. Shown (from left to right), are the output of the first layer of wav2vec 2.0, which contains speaker information, d-vector extracted from the speaker encoder used by AUTOVC, and the x-vector extracted by training the ECAPA-TDNN, which shows that the embedding performance of the x-vector extracted by training the ECAPA-TDNN model is the best.

The objective of the speaker encoder is to satisfy the following two conditions. 1) The speaker embeddings of the same speaker must be the same regardless of the utterance content, and 2) speaker embeddings of different speakers cannot be the same regardless of the utterance content.

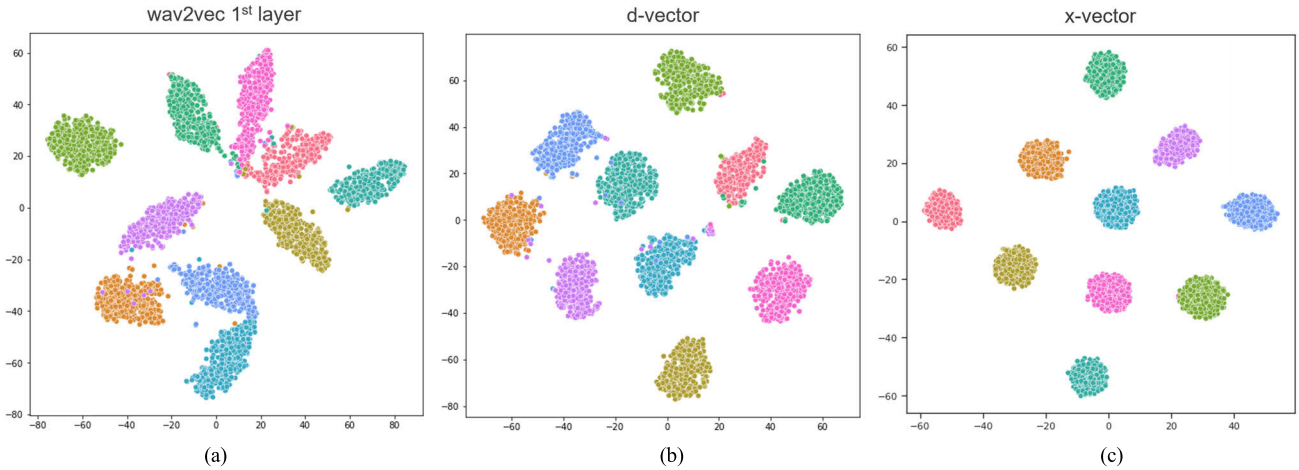

FIGURE 3. Comparison of visualization of speaker embedding using TSNE. The figure shows (from left to right) the interspeaker distribution of the first layer of wav2vec 2.0, d-vector, and x-vector. The embedding results for 10 unseen speakers show that the embedding results of the x-vector are much better than those of wav2vec 2.0 or d-vector.

Therefore, the speaker encoder must satisfy these two equations (see (5)), which is an important assumption in voice conversion.

$$\begin{aligned} &1)\ if\ U_1 = U_2,\ E_s(X_1) = E_s(X_2), \\ &2)\ if\ U_1 \neq U_2,\ E_s(X_1) \neq E_s(X_2), \end{aligned} \tag{5}$$

where U_1, U_2 represents the different speakers, X_1, X_2 represents the utterances of different speakers. The linguistic information contained in X_1 and X_2 may either be the same or different. E_s represents the speaker encoder.

C. DECODER

The decoder restores the original speech signal using both linguistic and speaker information, which are the outputs of the content and speaker encoders. In addition to the linguistic and speaker information, which are essential for reconstruction, the energy information extracted from the source speech is concatenated and used for reconstruction. Energy was used as the value obtained by averaging the log Mel-Spectrogram of the input data along the frequency axis, and the data used were not perturbed. As shown in Fig. 4, the time–frequency structure of the Mel-Spectrogram is well preserved after it passes through the decoder by using additional energy information for reconstruction.

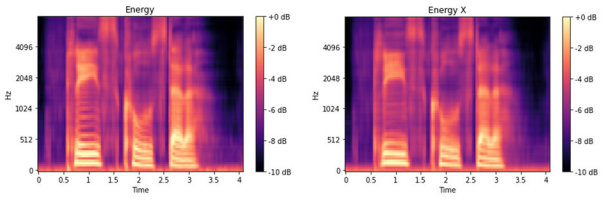

FIGURE 4. Output of Mel-Spectrogram with or without energy. On the left is the Mel-Spectrogram with energy information added, and on the right is the output of the Mel-Spectrogram without energy information.

The input data, which concatenate the three pieces of information, pass through a LSTM layer before and after they pass through three 5×1 convolutional layers. After passing through the last fully connected layer, the output was an 80-dimensional Mel-Spectrogram of the converted speech. If we arbitrarily call the converted speech $\hat{X}_1$, then $\hat{X}_1$ is used again as the input to the convolution layers, passing through five additional convolution layers. The converted voice that has passed through additional convolution layers is called $\hat{X}_2$. The composite speech (sum of $\hat{X}_1$ and $\hat{X}_2$) is the final converted speech. In this case, the convolution layers after the fully connected layer use a hyperbolic tangent as the activation function.

D. VOCODER

A HiFi-GAN was used as the vocoder to convert the Mel-Spectrogram back into a waveform [55].

E. TRAINING

The purpose of the content encoder, speaker encoder, and decoder can be expressed as

$$C_1 = E_c\left(\tilde{X}_1\right),\ S_1 = E_s(X_1),\ \hat{X}_{1\to 1} = D(C_1, S_1, E), \tag{6}$$

where X_1 represents the original voice of the source speaker used as input data and $\tilde{X}_1$ is the input data with the applied perturbation. E_c and E_s are the content and speaker encoders, respectively, and the resulting C_1 and S_1 are the content and speaker embedding information, respectively. $D(\cdot, \cdot, \cdot)$ denotes the decoder, and E refers to energy information.

Finally, the goal of learning is to use the information from the source speaker to restore the original data to the same level as it was before the application of perturbation through the

decoder. Therefore, we can formally express this as

$$L_{recon} = \mathbb{E}\left[\left\|\hat{X}_{1\to 1} - X_1\right\|_2^2\right]. \tag{7}$$

Hence, the overall loss used for learning was equal to (8) by adding (4) and (7). If idealized voice conversion is performed, the result follows the data distribution as shown in (9). At this point, the values of λ and μ in (8), are set to two and one, respectively. In (8), L_{recon1} is the loss of the first output $\hat{X}_1$, and L_{recon2} is the loss of the final output $\hat{X}_2$. In (9), U and Z denote the speaker and content of the utterance, respectively.

$$\min_{E_c(.),D_{(.,.,.)}} L = \lambda(L_{recon1} + L_{recon2}) + \mu L_{content}, \tag{8}$$

$$p\hat{x}_{1\to 2}(\cdot|U_2 = u_2, Z_1 = z_1) = px(\cdot|U = u_2, Z = z_1). \tag{9}$$

IV. EXPERIMENTS

To train Perturbation AUTOVC, we used the VCTK dataset, which contained approximately 44 hours of English speech from 109 speakers [56]. The VCTK dataset consists of English native speakers with diverse accents (i.e., American, Scottish, Irish etc.), and the gender distribution is balanced. The number of utterances for each speaker is approximately around 400. We used 80% of the total speakers for training (20% for evaluation), and we preserved 10 utterances from each speaker for evaluation purposes, which were not used in the training.

The input data used comprised a waveform sampled at a rate of 22.05 kHz, which was converted into an 80-dimensional Mel-Spectrogram. The fast Fourier transformation size, window size, and hop length, for the Mel-Spectrogram conversion were set to 1024, 1024, and 256, respectively. The converted Mel-Spectrograms were randomly truncated or padded such that they were all 128 frames long and used for training. The batch size was set to two, the learning rate was 0.0001, and ADAM was used as the optimizer. At this time, β_1 and β_2 of the ADAM optimizer were set to 0.5 and 0.9, respectively, and the λ and μ values were set to two and one, respectively. The optimizer's weight decay was set to 0, and regularization techniques such as L1 or L2 were not employed. The training iterations were set to 100,000, and the final reconstruction loss (see (7)) converged to around 0.1 at approximately 50,000 iterations. We trained using a single GPU on an A10 server, and the memory requirement is about 2800MiB. The Real Time Factor (RTF) for inference time is 1.58s.

In this study, we used three objective evaluation metrics and one subjective evaluation metric: Objective metrics, including CER, Cosine similarity, and STOI, along with the subjective metric, neural-MOS [57], [58], [59]. By measuring objective metrics such as CER and STOI, we can estimate the quality or naturalness of the converted speech, and by measuring cosine similarity, we can estimate speaker similarity.

A. CER

The CER (%) indicates the percentage of recognized character errors between the converted speech and source speech; the lower the CER (%) value is, the more effectively the converted speech preserves the linguistic information of the source speech. In this study, we used the ASR model of wav2vec2-base-960h and calculated CER (%) using the Levenshtein distance algorithm [60].

B. COSINE SIMILARITY (COSINE SCORE)

Cosine similarity is an indicator that can measure the similarity between two vectors using the cosine angle. Cosine similarity is a value that ranges between -1 and 1; the closer it is to one, the higher the similarity between the two vectors. We used this to measure the similarity between the speaker embedding of the target speaker extracted by the speaker encoder and the speaker embedding extracted from the converted speech as a metric to evaluate how closely a given speech was converted to the target speaker.

C. STOI

We measured the short-time objective intelligibility (STOI) of the converted speech for speech quality evaluation. STOI is an indicator of speech intelligibility, ranging from 0 to 1; closer to 1 indicates that the information contained in the original speech is transferred to the converted speech without loss. To measure the STOI of the converted speech, a sampling rate of 22.05kHz was used.

D. NEURAL MOS PREDICTOR

We also evaluated the naturalness of the converted speech using the neural MOS predictor (SSL-MOS), which is the baseline system in VoiceMOS Challenge 2022 [61], [62]. It is an evaluation metric used for MOS measurement in the Singing Voice Conversion Challenge 2023 [63]. SSL-MOS is a model that generalizes Mean Opinion Score (MOS) prediction performance on different listening test data in zero-shot and fine-tuning settings. It utilizes a MOS prediction network, including MOSNet and self-supervised speech models like wav2vec2 [64]. SSL-MOS finds that when wav2vec2 models are fine-tuned for MOS prediction, they exhibit good generalization capabilities, even in challenging zero-shot cases (out-of-domain data). Furthermore, it has been observed that fine-tuning on in-domain data can improve MOS prediction performance.

For comparison purposes, we choose recently proposed voice conversion models capable of one-shot many-to-many voice conversion.

E. AUTOVC

AUTOVC is an autoencoder-based voice conversion model with carefully designed bottleneck dimensions, and it is the first model to perform zero-shot voice conversion.

F. AGAIN-VC

AGAIN-VC is an improved version of the conventional autoencoder-based AdaIN-VC, utilizing activation guidance and adaptive instance normalization [65].

G. BNE-PPG-VC

PPG-VC combines a bottleneck feature extractor (BNE) with a seq2seq synthesis module [66].

H. VQMIVC

VQMIVC is a method that employs vector quantization in content encoding [67].

The voice conversion samples used in the evaluation were generated by a pre-trained VC model provided in the official GitHub repository. For a fair comparison with AUTOVC, the d-vector used as speaker information in the original AUTOVC was replaced by the x-vector used in this study, and the vocoder used to restore speech was replaced by WaveNet used in the original AUTOVC in the HiFi-GAN used in this study. Except for the PPG-VC model, all other models were trained on the VCTK dataset, while the PPG-VC model was trained on the large-scale dataset (VCTK+LibriTTS). For evaluation, we randomly selected 10 female and 10 male speakers from the VCTK dataset. For each speaker, we used 10 utterances that were not used in training for voice conversion. Voice conversion was performed for all possible cases and the performance was measured for a total of 4,000 samples ($_{20}\Pi_2 \times 100$).

In Table 1, it can be observed that the Perturbation AUTOVC model outperforms other models in objective metrics. For instance, in terms of CER (%), Perturbation AUTOVC achieved the lowest error rate at 6.8%, while other models exhibited relatively higher error rates of approximately 11%, 21%, and 31% each. Furthermore, when measuring the speaker similarity of the converted speech using Cosine Similarity, Perturbation AUTOVC also showed the highest similarity score at 0.58, while other models displayed relatively lower similarity scores. In STOI scores as well, Perturbation AUTOVC achieves the highest score of 0.82, while other models, excluding AUTOVC, exhibit relatively lower performance.

TABLE 1. Objective evaluation (CER, Cosine Score, STOI) of VC models on the VCTK dataset.

LABEL	CER (%)	Cosine	STOI
AGAIN-VC	31.0	0.43	0.22
VQMIVC	21.1	0.41	0.28
BNE-PPG-VC	11.4	0.37	0.64
AUTOVC	11.3	0.42	0.80
PERTURBATION-AUTOVC	**6.8**	**0.58**	**0.82**

In Table 2, it is observed that the Perturbation AUTOVC model outperforms other models in subjective metrics. Ground-truth is the average of the MOS scores measured using the source speaker's speech. Perturbation AUTOVC achieved an MOS score of 3.82 points, which is a 0.5-point difference from the Ground-truth of 4.31. Compared to other models, it demonstrates the closest approximation to the Ground-truth MOS results.

TABLE 2. Subjective evaluation (neural MOS) of VC models on the VCTK dataset.

LABEL	MOS
AGAIN-VC	2.18
VQMIVC	3.12
BNE-PPG-VC	3.54
AUTOVC	3.75
PERTURBATION-AUTOVC	**3.82**
Ground-truth	4.31

I. ZERO-SHOT VOICE CONVERSION

Finally, we performed zero-shot voice conversion with a speaker not used in training and evaluated it. For zero-shot voice conversion, we utilized LibriTTS, a dataset that was not used at all during training [68]. LibriTTS is a large-scale multi-speaker English speech dataset derived from the LibriSpeech dataset [69]. We used the 'train-clean-100' dataset, which consists of a total of 257 speakers. From this dataset, we randomly selected 5 male and 5 female speakers, resulting in a total of 10 speakers. For each of these 10 speakers, we performed voice conversion using 5 utterances per speaker. BNE-PPG-VC was excluded from the zero-shot voice conversion performance comparison because it includes the LibriTTS dataset in its training data.

In Table 3, it can be observed that the Perturbation AUTOVC model demonstrates excellent performance even in zero-shot conversion. In terms of CER (%), Perturbation AUTOVC achieved the lowest error rate of 2.2%, while other models show higher error rates of approximately 14%, 27%, and 54%. Even though the LibriTTS dataset used for zero-shot conversion is entirely new data not included in the training set, Table 3 shows a lower CER compared to the VCTK dataset used for training. This can be attributed to the difference between the Ground-truth of the VCTK dataset in Table 2, which is 4.31, and the Ground-truth of the LibriTTS dataset in Table 4, which is 4.74. We speculate that LibriTTS speech has higher quality compared to VCTK speech, which likely influenced the CER (%). Within the AUTOVC results, CER represents a very high error rate of 54.3%. This high error rate is attributed to the bottleneck

TABLE 3. Objective evaluation (CER, Cosine Score, STOI) of VC models in zero-shot voice conversion.

LABEL	CER (%)	Cosine	STOI
AGAIN-VC	14.1	**0.47**	0.43
VQMIVC	27.4	0.15	0.64
AUTOVC	54.3	0.32	0.36
PERTURBATION-AUTOVC	**2.2**	0.42	**0.80**

TABLE 4. Subjective evaluation (neural MOS) of VC models in zero-shot voice conversion.

LABEL	MOS
AGAIN-VC	1.83
VQMIVC	2.05
AUTOVC	1.66
PERTURBATION-AUTOVC	**3.56**
Ground-truth	4.74

dimension, which is carefully designed to fit the VCTK dataset, may not be well-suitable for disentangling linguistic information from speaker information in the speech of LibriTTS. In terms of cosine similarity, AGAIN-VC scores 0.47 and Perturbation AUTOVC scores 0.42, which outperforms other models. In the STOI score comparison, the Perturbation AUTOVC achieved a score of 0.80, significantly outperforming the performance of other models. In these three-objective metrics, the proposed method, Perturbation AUTOVC, shows remarkable performance with a very low CER of 2.2% and a high STOI score of 0.80. Compared to AGAIN-VC, which achieved the highest cosine similarity score of 0.47, the Perturbation AUTOVC shows a difference of 0.05. Therefore, considering all three evaluation indicators in Table 3, the proposed method outperforms other models in a zero-shot environment.

In Table 4, we can observe that the Perturbation AUTOVC model outperforms other models in subjective evaluations even in zero-shot conversion. Ground-truth is the average of the MOS scores measured using the source speaker's speech from the LibriTTS dataset. Perturbation AUTOVC achieved the highest MOS score of 3.56 points. Comparing Table 2 and Table 4, it is evident that the other models MOS scores when performing zero-shot in Table 4 are significantly lower, around 2 points, compared to MOS scores above 3 points with the VCTK dataset. However, the proposed approach, Perturbation AUTOVC, demonstrates robust performance with a difference of only about 0.3 points between its performance on the VCTK dataset and zero-shot performance on unseen datasets. Based on the zero-shot performance results, Perturbation AUTOVC demonstrates its ability to perform robust voice conversion on new datasets.

V. CONCLUSION

In this study, we proposed Perturbation AUTOVC, a nonparallel voice conversion method that utilizes only perturbation and autoencoder loss. The proposed method demonstrated the effectiveness of voice conversion by combining a feature extraction method using information perturbation and a self-reconstruction method using an autoencoder structure. In terms of CER (%), cosine similarity, STOI evaluations, and MOS results, Perturbation AUTOVC outperformed other voice conversion techniques in all aspects, particularly demonstrating very high performance in zero-shot voice conversion. The information contained in speech includes not only speaker and linguistic information but also long-term and acoustic features such as prosody and reverberation. However, in this study, due to the binarization of speech information into speaker and linguistic information, limitations may arise from this approach. In future research, we will introduce methods such as using additional encoders to process more diverse information.

REFERENCES

[1] B. Sisman, J. Yamagishi, S. King, and H. Li, "An overview of voice conversion and its challenges: From statistical modeling to deep learning," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 29, pp. 132–157, 2021.

[2] S. H. Mohammadi and A. Kain, "An overview of voice conversion systems," *Speech Commun.*, vol. 88, pp. 65–82, Apr. 2017.

[3] G. P. Prajapati, D. K. Singh, P. P. Amin, and H. A. Patil, "Voice privacy using CycleGAN and time-scale modification," *Comput. Speech Lang.*, vol. 74, Jul. 2022, Art. no. 101353.

[4] A. Přibilová and J. Přibil, "Non-linear frequency scale mapping for voice conversion in text-to-speech system with cepstral description," *Speech Commun.*, vol. 48, no. 12, pp. 1691–1703, Dec. 2006.

[5] L. Xue, S. Pan, L. He, L. Xie, and F. K. Soong, "Cycle consistent network for end-to-end style transfer TTS training," *Neural Netw.*, vol. 140, pp. 223–236, Aug. 2021.

[6] Y. Stylianou, O. Cappe, and E. Moulines, "Continuous probabilistic transform for voice conversion," *IEEE Trans. Speech Audio Process.*, vol. 6, no. 2, pp. 131–142, Mar. 1998.

[7] T. Toda, A. W. Black, and K. Tokuda, "Voice conversion based on maximum-likelihood estimation of spectral parameter trajectory," *IEEE Trans. Audio, Speech Language Process.*, vol. 15, no. 8, pp. 2222–2235, Nov. 2007.

[8] L. Sun, S. Kang, K. Li, and H. Meng, "Voice conversion using deep bidirectional long short-term memory based recurrent neural networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2015, pp. 4869–4873.

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

[10] Z. Wu, T. Virtanen, E. S. Chng, and H. Li, "Exemplar-based sparse representation with residual compensation for voice conversion," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 22, no. 10, pp. 1506–1521, Oct. 2014.

[11] X. T. Chen and L. H. Zhang, "High-quality voice conversion system based on GMM statistical parameters and RBF neural network," *J. China Univ. Posts Telecommun.*, vol. 21, no. 5, pp. 68–75, Oct. 2014.

[12] N. J. Shah and H. A. Patil, "A novel approach to remove outliers for parallel voice conversion," *Comput. Speech Lang.*, vol. 58, pp. 127–152, Nov. 2019.

[13] M. Ghorbandoost, A. Sayadiyan, M. Ahangar, H. Sheikhzadeh, A. S. Shahrebabaki, and J. Amini, "Voice conversion based on feature combination with limited training data," *Speech Commun.*, vol. 67, pp. 113–128, Mar. 2015.

[14] M. Jafaryani, H. Sheikhzadeh, and V. Pourahmadi, "Parallel voice conversion with limited training data using stochastic variational deep kernel learning," *Eng. Appl. Artif. Intell.*, vol. 115, Oct. 2022, Art. no. 105279.

[15] H. Du, X. Tian, L. Xie, and H. Li, "Factorized WaveNet for voice conversion with limited data," *Speech Commun.*, vol. 130, pp. 45–54, Jun. 2021.

[16] B. Chen, Z. Xu, and K. Yu, "Data augmentation based non-parallel voice conversion with frame-level speaker disentangler," *Speech Commun.*, vol. 136, pp. 14–22, Jan. 2022.

[17] M. S. Al-Radhi, T. G. Csapó, and G. Németh, "Continuous vocoder applied in deep neural network based voice conversion," *Multimedia Tools Appl.*, vol. 78, no. 23, pp. 33549–33572, Dec. 2019.

[18] F.-L. Xie, F. K. Soong, and H. Li, "Voice conversion with Si-DNN and KL divergence based mapping without parallel training data," *Speech Commun.*, vol. 106, pp. 57–67, Jan. 2019.

[19] F. Liu, H. Wang, Y. Ke, and C. Zheng, "One-shot voice conversion using a combination of U2-net and vector quantization," *Appl. Acoust.*, vol. 199, Oct. 2022, Art. no. 109014.

[20] H. Du, L. Xie, and H. Li, "Noise-robust voice conversion with domain adversarial training," *Neural Netw.*, vol. 148, pp. 74–84, Apr. 2022.

[21] Z. Cai, Y. Yang, and M. Li, "Cross-lingual multi-speaker speech synthesis with limited bilingual training data," *Comput. Speech Lang.*, vol. 77, Jan. 2023, Art. no. 101427.

[22] D. Popek and U. Markowska-Kaczmar, "Utterance style transfer using deep models," *Proc. Comput. Sci.*, vol. 192, pp. 2132–2141, Jan. 2021.

[23] L. Sun, K. Li, H. Wang, S. Kang, and H. Meng, "Phonetic posteriorgrams for many-to-one voice conversion without parallel data training," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jul. 2016, pp. 1–6.

[24] W.-Z. Zheng, J.-Y. Han, C.-K. Lee, Y.-Y. Lin, S.-H. Chang, and Y.-H. Lai, "Phonetic posteriorgram-based voice conversion system to improve speech intelligibility of dysarthric patients," *Comput. Methods Programs Biomed.*, vol. 215, Mar. 2022, Art. no. 106602.

[25] M. Zhang, B. Sisman, L. Zhao, and H. Li, "DeepConversion: Voice conversion with limited parallel training data," *Speech Commun.*, vol. 122, pp. 31–43, Sep. 2020.

[26] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2242–2251.

[27] T. Kaneko and H. Kameoka, "CycleGAN-VC: Non-parallel voice conversion using cycle-consistent adversarial networks," in *Proc. 26th Eur. Signal Process. Conf. (EUSIPCO)*, Sep. 2018, pp. 2100–2104.

[28] T. Kaneko, H. Kameoka, K. Tanaka, and N. Hojo, "Cyclegan-VC2: Improved cyclegan-based non-parallel voice conversion," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2019, pp. 6820–6824.

[29] T. Kaneko, H. Kameoka, K. Tanaka, and N. Hojo, "CycleGAN-VC3: Examining and improving CycleGAN-VCs for mel-spectrogram conversion," in *Proc. Interspeech*, Oct. 2020.

[30] T. Kaneko, H. Kameoka, K. Tanaka, and N. Hojo, "Maskcyclegan-VC: Learning non-parallel voice conversion with filling in frames," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Jun. 2021, pp. 5919–5923.

[31] 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 *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8789–8797.

[32] H. Kameoka, T. Kaneko, K. Tanaka, and N. Hojo, "StarGAN-VC: Non-parallel many-to-many voice conversion using star generative adversarial networks," in *Proc. IEEE Spoken Lang. Technol. Workshop (SLT)*, Dec. 2018, pp. 266–273.

[33] T. Kaneko, H. Kameoka, K. Tanaka, and N. Hojo, "StarGAN-VC2: Rethinking conditional methods for StarGAN-based voice conversion," in *Proc. Interspeech*, Sep. 2019, pp. 679–683.

[34] H. S. Choi, J. Lee, W. Kim, J. Lee, H. Heo, and K. Lee, "Neural analysis and synthesis: Reconstructing speech from self-supervised representations," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, vol. 34, 2021, pp. 16251–16265.

[35] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "Wav2vec 2.0: A framework for self-supervised learning of speech representations," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, vol. 33, 2020, pp. 12449–12460.

[36] Z. Bai and X.-L. Zhang, "Speaker recognition based on deep learning: An overview," *Neural Netw.*, vol. 140, pp. 65–99, Aug. 2021.

[37] A. Nagrani, J. S. Chung, W. Xie, and A. Zisserman, "Voxceleb: Large-scale speaker verification in the wild," *Comput. Speech Lang.*, vol. 60, Mar. 2020, Art. no. 101027.

[38] J. S. Chung, A. Nagrani, and A. Zisserman, "VoxCeleb2: Deep speaker recognition," in *Proc. Interspeech*, Sep. 2018.

[39] Y. Jia, Y. Zhang, R. Weiss, Q. Wang, J. Shen, F. Ren, Z. Chen, P. Nguyen, R. Pang, I. Lopez-Moreno, and Y. Wu, "Transfer learning from speaker verification to multispeaker text-to-speech synthesis," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, vol. 31, 2018, pp. 4485–4495.

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

[41] N. Jaitly and G. E. Hinton, "Vocal tract length perturbation (VTLP) improves speech recognition," in *Proc. ICML Workshop Deep Learn. Audio, Speech Lang.*, vol. 117, Jun. 2013, p. 21.

[42] Z. Ning, Q. Xie, P. Zhu, Z. Wang, L. Xue, J. Yao, L. Xie, and M. Bi, "Expressive-VC: Highly expressive voice conversion with attention fusion of bottleneck and perturbation features," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Jun. 2023, pp. 1–5.

[43] T. Toda, L.-H. Chen, D. Saito, F. Villavicencio, M. Wester, Z. Wu, and J. Yamagishi, "The voice conversion challenge 2016," in *Proc. Interspeech*, Sep. 2016, pp. 1632–1636.

[44] W. Fedus, I. Goodfellow, and A. M. Dai, "MaskGAN: Better text generation via filling in the _____," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2018.

[45] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. Annu. Conf. North Amer. Chapter Assoc. Compitational Linguistics (NAACL)*, 2019, pp. 4171–4186.

[46] C. Fu, C. Liu, C. T. Ishi, and H. Ishiguro, "An improved CycleGAN-based emotional voice conversion model by augmenting temporal dependency with a transformer," *Speech Commun.*, vol. 144, pp. 110–121, Oct. 2022.

[47] J. Shah, Y. K. Singla, C. Chen, and R. R. Shah, "What all do audio transformer models hear? Probing acoustic representations for language delivery and its structure," 2021, *arXiv:2101.00387*.

[48] L. Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, p. 11, Nov. 2008.

[49] Z. Fan, M. Li, S. Zhou, and B. Xu, "Exploring wav2vec 2.0 on speaker verification and language identification," 2020, *arXiv:2012.06185*.

[50] P. Boersma and V. Van Heuven, "Speak and unspeak with PRAAT," *Glot Int.*, vol. 5, nos. 9–10, pp. 341–347, 2001.

[51] B. Desplanques, J. Thienpondt, and K. Demuynck, "ECAPA-TDNN: Emphasized channel attention, propagation and aggregation in TDNN based speaker verification," in *Proc. Interspeech*, Oct. 2020, pp. 3830–3834.

[52] W. Xie, A. Nagrani, J. S. Chung, and A. Zisserman, "Utterance-level aggregation for speaker recognition in the wild," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2019, pp. 5791–5795.

[53] Z. Gao, Y. Song, I. McLoughlin, P. Li, Y. Jiang, and L.-R. Dai, "Improving aggregation and loss function for better embedding learning in end-to-end speaker verification system," in *Proc. Interspeech*, Sep. 2019, pp. 361–365.

[54] S. H. Mun, J.-W. Jung, M. H. Han, and N. S. Kim, "Frequency and multi-scale selective kernel attention for speaker verification," 2022, *arXiv:2204.01005*.

[55] J. Kong, J. Kim, and J. Bae, "HiFi-GAN: Generative adversarial networks for efficient and high fidelity speech synthesis," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 17022–17033.

[56] C. Veaux, J. Yamagishi, and K. MacDonald, "Superseded-CSTR VCTK-corpus: English multi-speaker corpus for CSTR voice cloning toolkit," Centre Speech Technol. Res. (CSTR), Univ. of Edinburgh, Edinburgh, U.K., Tech. Rep., 2016.

[57] I. S. MacKenzie and R. W. Soukoreff, "A character-level error analysis technique for evaluating text entry methods," in *Proc. 2nd Nordic Conf. Hum.-Comput. Interact.*, Oct. 2002, pp. 243–246.

[58] K. K. George, C. S. Kumar, S. Sivadas, K. I. Ramachandran, and A. Panda, "Analysis of cosine distance features for speaker verification," *Pattern Recognit. Lett.*, vol. 112, pp. 285–289, Sep. 2018.

[59] C. H. Taal, R. C. Hendriks, R. Heusdens, and J. Jensen, "A short-time objective intelligibility measure for time-frequency weighted noisy speech," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2010, pp. 4214–4217.

[60] S. Zhang, Y. Hu, and G. Bian, "Research on string similarity algorithm based on levenshtein distance," in *Proc. IEEE 2nd Adv. Inf. Technol., Electron. Autom. Control Conf. (IAEAC)*, Mar. 2017, pp. 2247–2251.

[61] E. Cooper, W.-C. Huang, T. Toda, and J. Yamagishi, "Generalization ability of MOS prediction networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2022, pp. 8442–8446.

[62] W.-C. Huang, E. Cooper, Y. Tsao, H.-M. Wang, T. Toda, and J. Yamagishi, "The VoiceMOS challenge 2022," 2022, *arXiv:2203.11389*.

[63] W.-C. Huang, L. P. Violeta, S. Liu, J. Shi, and T. Toda, "The singing voice conversion challenge 2023," 2023, *arXiv:2306.14422*.

[64] C.-C. Lo, S.-W. Fu, W.-C. Huang, X. Wang, J. Yamagishi, Y. Tsao, and H.-M. Wang, "MOSNet: Deep learning-based objective assessment for voice conversion," in *Proc. Interspeech*, Sep. 2019, pp. 1541–1545.

[65] 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 *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Jun. 2021, pp. 5954–5958.

[66] 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 Trans. Audio, Speech, Language Process.*, vol. 29, pp. 1717–1728, 2021.
[67] 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," in *Proc. Interspeech*, 2021, pp. 1344–1348.
[68] H. Zen, V. Dang, R. Clark, Y. Zhang, R. J. Weiss, Y. Jia, Z. Chen, and Y. Wu, "LibriTTS: A corpus derived from LibriSpeech for text-to-speech," in *Proc. Interspeech*, Sep. 2019, pp. 1526–1530.
[69] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An ASR corpus based on public domain audio books," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2015, pp. 5206–5210.

HWA-YOUNG PARK is currently pursuing the degree with the Department of Computer Engineering, Chosun University, Gwangju. She is an Undergraduate Research Student with the Advanced Multimedia Computing Laboratory (AMCL). Her current research interests include voice conversion and deep learning.

YOUNG HAN LEE received the B.S. degree in electronics engineering from Gwangwoon University, South Korea, in 2005, and the M.S. and Ph.D. degrees in electrical engineering and computer science from the Gwangju Institute of Science and Technology (GIST), in 2007 and 2011, respectively. From 2011 to 2014, he was a Senior Researcher with the LG Advanced Research Institute, South Korea. Since 2015, he has been a Principal Researcher with the Department of Intelligent Information Research, Korea Electronics Technology Institute (KETI). His current research interests include speech and audio signal generation, and multimodal processing.

CHANJUN CHUN (Member, IEEE) received the B.S. degree in electronics engineering from the Korea University of Technology and Education, South Korea, in 2009, and the M.S. and Ph.D. degrees in electrical engineering and computer science from the Gwangju Institute of Science and Technology (GIST), in 2011 and 2017, respectively. From 2017 to 2021, he was a Senior Researcher with the Korea Institute of Civil Engineering and Building Technology (KICT), South Korea. Since March 2021, he has been an Assistant Professor with the Department of Computer Engineering, Chosun University, Gwangju, South Korea. His current research interests include speech and audio signal processing, and deep learning.