# Pureformer-VC: Non-parallel Voice Conversion with Pure Stylized Transformer Blocks and Triplet Discriminative Training

1st Wenhan Yao
*School of Computer Science*
*Xiangtan University*
Xiangtan, China
wenhanyao@smail.xtu.edu.cn

2nd Fen Xiao
*School of Computer Science*
*Xiangtan University*
Xiangtan, China
xiaof@xtu.edu.cn

3rd Xiarun Chen
*School of Software Microelectronics*
*Peking University*
Beijing, China
xiar_c@pku.edu.cn

4th Jia Liu
*School of Software Microelectronics*
*Peking University*
Beijing, China
2201120008@stu.edu.cn

5th YongQiang He
*School of Software Microelectronics*
*Peking University*
Beijing, China
heyongqiang@stu.pku.edu.cn

6th Weiping Wen
*School of Software Microelectronics*
*Peking University*
Beijing, China
weipingwen@pku.edu.cn

2025 International Joint Conference on Neural Networks (IJCNN) | 979-8-3315-1042-8/25/$31.00 ©2025 IEEE | DOI: 10.1109/IJCNN64981.2025.11229291

***Abstract*—As a foundational technology for intelligent human-computer interaction, voice conversion (VC) seeks to transform speech from any source timbre into any target timbre. Traditional voice conversion methods based on Generative Adversarial Networks (GANs) encounter significant challenges in precisely encoding diverse speech elements and effectively synthesising these elements into natural-sounding converted speech. To overcome these limitations, we introduce Pureformer-VC, an encoder-decoder framework that utilizes Conformer blocks to build a disentangled encoder and employs Zipformer blocks to create a style transfer decoder. We adopt a variational decoupled training approach to isolate speech components using a Variational Autoencoder (VAE), complemented by triplet discriminative training to enhance the speaker's discriminative capabilities. Furthermore, we incorporate the Attention Style Transfer Mechanism (ASTM) with Zipformer's shared weights to improve the style transfer performance in the decoder. We conducted experiments on two multi-speaker datasets. The experimental results demonstrate that the proposed model achieves comparable subjective evaluation scores while significantly enhancing objective metrics compared to existing approaches in many-to-many and many-to-one VC scenarios.**

***Index Terms*—VC, VAE, Styleformer, Conformer, Zipformer.**

## I. Introduction

Voice conversion (VC) seeks to transform the speaker's timbre in speech to match that of a target speaker while preserving the original content. The task is typically text-independent. Traditional parallel VC research has primarily focused on feature matching methods [1]–[5], with the corpus consisting of paired utterances that have identical linguistic content. Consequently, these methods struggle to address the challenge of converting between a wide range of different timbres and generating natural speech quality. Recently, researchers have increasingly concentrated on non-parallel VC trained on multi-speaker datasets featuring randomly spoken utterances. Non-parallel VC encompasses many-to-many and one-to-many VC tasks, designed to generate diverse timbres or a specific target timbre from various source timbres. In this scenario, the target timbre only minimally participates or does not engage in the VC model training.

Drawing inspiration from the concept of image style transfer in computer vision, generative adversarial networks (GANs) have surfaced as a formidable tool for achieving non-parallel voice conversion (VC). Several GAN-based VC methods have been proposed [6]–[11], which do not require explicit parallel target utterances for training. Instead, a discriminator assesses whether a GAN-based VC model produces speech that embodies the target voice characteristics. Consequently, GAN-based VC models learn to convert voice across trained timbres, resulting in limited timbre targets. Within these methods, speaker encoders and style transfer functions play an essential role. They assist the generator in understanding transformation relationships between various speaker domains, which is achieved through the integration of style transfer modules into the generators, such as Adaptive Instance Normalization (AdaIN) [12] and Weight Adaptive Instance Normalization (WadaIN) [13].

However, training GAN models remains challenging due to issues with convergence and sensitivity to dataset imbalances. In recent years, flow-based VC methods (such as SoftVC-VITS [14] and YourTTS [15]), KNN-based VC approaches (like KNN-VC [16] and its derivative project RVC[1]), and Generative Large Language Model based VC (GLLM-VC), including GPT-SoVITS[2], have significantly improved audio quality and training stability. Flow-matching VC and some

[1] https://github.com/RVC-Project/Retrieval-based-Voice-Conversion-WebUI
[2] https://github.com/RVC-Boss/GPT-SoVITS

GLLM-VC techniques utilize invertible flow architectures to transform the timbre in either the frequency domain of speech or the discrete speech unit. The KNN-VC method substitutes the source content units with the nearest content units from the target timbre match set. These high-quality VC methods require deep speech unit learning with a multi-speaker corpus pre-training, which are time-consuming in practice.

Considering that speech can be broken down into multiple components [17] (e.g., timbre, pitch, content, and rhythm), disentanglement-based VC appears to be a promising approach. This framework enables neural networks to develop distinct representations of each speech component using several encoders and a decoder [17], [18]. During training, each encoder analyzes the corresponding spectrogram to create independent representations of the speech components. The decoder then combines these components to reconstruct the original speech. However, current methods—such as the forced decomposition in SpeechSplit [17], [18], INVC [19], and the information bottleneck strategy in AutoVC [20]—do not ensure perfect disentanglement or high-quality reconstruction.

*So, how should an efficient encoder-decoder framework be constructed for VC tasks?*

We propose that an effective and practical disentangled voice conversion framework, based on an encoder-decoder architecture, must be fundamentally grounded in three essential principles: (1) encoders and decoders with distinct roles, (2) an optimization objective that enhances representational discriminability, and (3) an efficient style transfer module within the decoder to merge speech components and enable precise speech reconstruction. In recent years, the most effective architectures in the field of speech model backbones have been several enhanced transformer-based networks, such as Conformer [21], Paraformer [22], and Zipformer [23]. These models have shown exceptional sequence modeling capabilities and have achieved significant success in applications such as automatic speech recognition [24], speaker verification [25], [26], and speech enhancement [27], [28]. Consequently, we believe that constructing a VC framework utilizing Transformer-based networks is feasible.

Building on the previous discussion, we present **Pureformer-VC**[3] as a comprehensive solution for a practical VC framework with three technical approaches. For the first principle, we design a specialized content encoder that combines Conformer blocks with IN operations. This structure enhances the model's ability to represent linguistic information through normalized distributions while filtering out speaker characteristics. For the decoder, we utilize Zipformer blocks, which have shown exceptional performance in speech acoustic modeling tasks, to ensure high-quality synthesis output. For the second principle, we integrate the Attention Style Transfer Mechanism (ASTM) ootnotewu2021styleformer within the Zipformer blocks, effectively incorporating speaker information into generated speech. The speaker encoder is likewise constructed using Conformer blocks but omits IN to prevent the potential filtering of speaker information. For the final principle, we introduce a triplet loss [29] alongside the reconstruction loss, allowing the model to learn and maintain distinct distances between utterances of different timbres. In summary, this paper's main contributions are as follows.

- We proposed a one-shot, many-to-many VC framework called Pureformer-VC. The key modules are constructed using advanced speech encoding blocks, which assist in preserving the reconstruction quality.
- To enhance style transfer, the shared weights in Zipformer are implemented using the ASTM in Styleformer.
- We conducted one-shot and many-to-many voice conversion experiments on the VCTK and AISHELL-3 datasets. The evaluation results indicate that our proposed method achieves comparable or even superior results in various voice conversion scenarios compared to existing methods.

## II. Related Work

### A. Voice Conversion

Voice conversion (VC) model training can be broadly categorized into parallel and non-parallel approaches. Early parallel methods relied on utterances with identical content but varying timbres to map features. Techniques such as Gaussian Mixture Models (GMM-VC) [1], Directional Kernel Partial Least Squares (DKPLS) [2], Vector Quantization-based VC (VQ-VC) [3], frequency warping [4], and Non-Negative Matrix Factorization (NMF) [5] were frequently employed. However, these approaches often yielded overly smooth outputs and demonstrated weak generative performance. Recent technological advancements have triggered a paradigm shift toward non-parallel voice conversion, which can be systematically classified into two primary research directions:

**Domain Transfer.** Domain transfer-based VC models, such as the series models of StarGAN-VC and CycleGAN-VC, treat each timbre as a domain and employ cyclic adversarial training to transform features across domains. This method facilitates the generation of more realistic and diverse speech outputs. A framework based on generative adversarial networks (GANs) has proven effective for non-parallel conversion.

**Information Disentanglement.** Information disentanglement seeks to break down speech into distinct components through an encoder, including content, timbre, pitch, and rhythm, thereby allowing flexible recombination by a decoder. Representative models, such as INVC, SpeechSplit, and MAIN-VC [30], utilize techniques like IN and mutual information estimation to achieve effective disentanglement and reconstruction.

**Generative VC.** We consider recent VC methods based on generative theory to be generative VC approaches, which include flow-based VC [14], [15], KNN-based VC [16], [31], and GLLM-VC (including GPT-SoVITS, etc.). In the voice conversion stage, flow-based models can invert the semantics of speech into textual information and then convert it back into speech with a new timbre. The KNN-based VC replaces the source speech's deep speech unit vectors with the given target

[3] https://github.com/ywh-my/PureformerVC

speech. The GLLM-VC models pretrain the discrete speech semantic representations using regression training and predict different speech timbres based on reference speech prompts.

This evolution from parallel to non-parallel methods emphasizes the shift toward models with enhanced flexibility, robustness, and generative capabilities in VC tasks. Formally, most non-parallel VC models can be represented as $y = G(x_s, x_y)$, where $x_s$ represents the speech providing the content and $x_y$ represents the speech supplying the speaker's voice characteristics.

### *B. Style Transfer Learning in VC*

Style transfer learning teaches VC models to merge various speech representations. Accordingly, the style transfer function accepts both source speaker-independent and target speaker-dependent representations. Chou et al. [19] were the first to discover that IN can filter out speaker information while retaining the source content from original utterances in INVC. Subsequently, the IN function found widespread application in GANs-based VC. [9], [10]. Furthermore, the WadaIN method implements affine operations on the convolutional kernel in Convolutional Neural Networks (CNNs), modifying the style of source data in WadaIN-VC [11] by convolving the source data. However, these models depend on CNN architectures and exhibit limited reconstruction capabilities. To leverage the self-attention mechanism in Transformers, the Attention-AdaIN-VC [32] incorporated the styleformer block within the CNN blocks, achieving improved VC performance. In styleformer [33], self-attention weights are stylized using speaker representations, successfully training an image style transfer model. In the styleformer's blocks, ASTM is commonly used to integrate individual embedding with self-attention layers. In this paper, we propose incorporating ASTM into our decoder as a style transfer module.

## III. Methodology

### *A. Overall Architecture*

The overall architecture of Pureformer-VC is depicted in Figure 1(a). Pureformer-VC consists of a content encoder, a decoder, a speaker encoder, and a vocoder. We utilized a pretrained Hifi-GAN generator [34] as the vocoder, which remains frozen during the training stage.

We set the input feature mel-spectrogram as $x \in X^{[L,D]}$, where L denotes the frame number and D represents the number of mel filters. The content encoder $E_c$ extracts the posterior variances of the content representation $r_m, r_s = E_c(x)$. The speaker encoder $E_s$ generates speaker embeddings as the timbre representation $s = E_s(x)$ from mel-spectrograms. The decoder $E_d$ takes the reparameterization variance and outputs the converted spectrogram $x_{dec}$, incorporating the style embedding for transfer. It is important to note that $e$ is a random variable that follows the standard normal distribution $N(0, 1)$.

### *B. Content Encoder with VAE Training*

The content encoder parameterizes and approximates the variational distribution of $q_\phi(z|x)$. Each Conformer block was constructed **with an IN function** and an AveragePooling1D layer following the convolution module to reduce the time dimension by half. There are four continuous blocks as shown in Figure 1 (a); thus, the output spectrogram's frame length is decreased by 16 times. Finally, the content encoder outputs the reparameterization of the content representation by two convolution layers:

$$r_m = Mean\_Conv(h_{enc}) \tag{1}$$

$$r_s = Std\_Conv(h_{enc}) \tag{2}$$

$$r_c = r_m + e * r_s \tag{3}$$

The $Mean\_Conv$ and $Std\_Conv$ denote the two convolution layers after the output $h_{enc}$ of Conformer blocks. Thus, the variable $r_c$ follows a normal distribution with mean $r_m$ and variance $r_s$. The reparameterization operation ensures that the speech latent variables input into the decoder remain typically distributed while also preserving the gradient propagation in the model. It turns out to optimize the Evidence Lower Bound (ELBO) [35] of $log(p(x))$:

$$L_{elbo} = E[logP_\theta(x|z)] - KL(q_\phi(z|x)||p(z)) \tag{4}$$

where $\phi$ denotes the encoder network and $\theta$ represents the decoder. The first term above is the reconstruction loss, while the second is the KullbackLeibler divergence between the approximate posterior and the prior. Thus, the VAE training loss can be summarized as:

$$L_{vae}(x, x_{dec}) = E\left[|x - x_{dec}|\right] + \tag{5}$$

$$0.5 \cdot E[r_c + r_m^2 - log(r_m^2) - 1] \tag{6}$$

The $r_c$ is derived from the input mel-spectrogram $x$.

### *C. ASTM in Decoder*

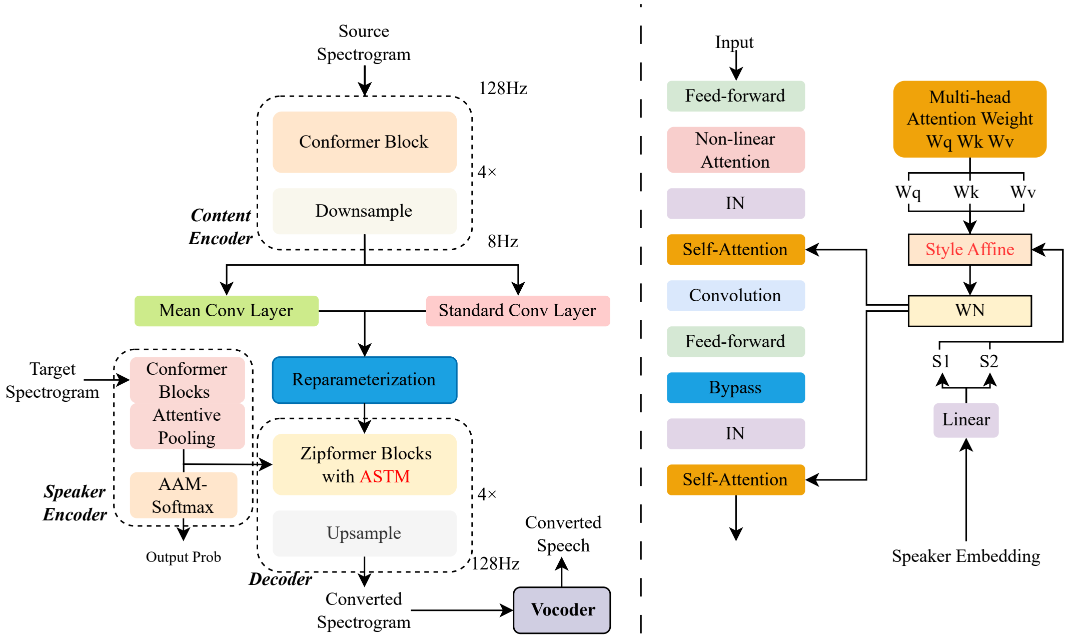
ASTM aids the decoder in learning the timbral characteristics of the target speech. We built the decoder using 4 Zipformer blocks with ASTM, as illustrated in Figure 1(b). The decoder merges the content and timbre representations of the speech.

To generate speech with diverse voice styles, we apply the ASTM to the weights in the self-attention mechanism of Zipformer blocks. During the model initialization phase, the ASTM initially sets some attention weights $w_q, w_k, w_v, w_u$. These weights are infused with the style characteristics of the split speaker embedding vector $s_1, s_2 = split(E_s(x))$ as follows:

$$w_q = w_q \cdot s_1 + s_2, w_k = w_k \cdot s_1 + s_2 \tag{7}$$

$$w_v = w_v \cdot s_1 + s_2, w_u = w_u \cdot s_1 + s_2 \tag{8}$$

Then, we apply weight normalization (WN) [36] to the weights to achieve improved convergence performance. WN



(a) Pureformer-VC

(b) A Zipfomer Block with ASTM

Fig. 1. The architecture of Pureformer-VC.

takes the weight $w$ and normalizes it at the output dimension $i, j$ as follows:

$$w'_{ij} = w_{ij} \cdot \frac{1}{\sqrt{w_{ij}^2}} \tag{9}$$

Using the WN operation, we scale the output of each weight $w \in \{w_q, w_k, w_v, w_u\}$ back to a unit standard deviation. The WN aids the model in accelerating training convergence following the attention calculation with stylized weights. Consequently, in the self-attention layers, the stylized attention is as follows:

$$x' = norm(s_1 \cdot x + s_2) \tag{10}$$

$$out = \frac{wn(w_q)x' \cdot (wn(w_k)x')^t}{\sqrt{d}} \cdot wn(w_k)x' + wn(w_u)x' \tag{11}$$

The $norm$ represents a non-parameterized layer normalization function. Additionally, $d$ indicates the output dimension for each weight, and $wn$ refers to the WN. The stylized attention output is connected residually.

### D. Speaker Encoder with AAM-Softmax Loss

The speaker encoder is designed to extract timbre representations from mel-spectrograms, allowing the model to capture speaker-specific features that are critical for VC. For its backbone, we use a structure made up of several Conformer blocks from MFA-Conformer [25]. Importantly, these Conformer blocks are configured **without the IN functions** to ensure the preservation of speaker-related information.

To further enhance the quality of the extracted embeddings, we incorporate the Additive Angular Margin Softmax (AAM-softmax) layer. This parameterized loss function optimizes the learning of compact and well-separated clusters in the embedding space for different speakers. By introducing a fixed angular margin between classes, the AAM-softmax layer encourages the encoder to produce discriminative and robust embeddings, thus improving the overall performance of speaker representation in the VC process.

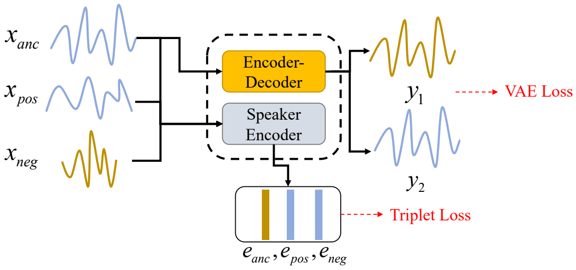


Fig. 2. The illustration of the training objective.

### E. Triplet loss and Data Sample Strategy

Considering the previous disentanglement-based VC models, both the source and target mel-spectrograms were identical during the training stage but differed during the inference stage. This discrepancy between training and inference diminishes the model's generalization capability and results in poor speech quality when the target spectrograms lack speaker information. To address this issue, we utilize the triplet loss, as shown in Figure 2, which is an unsupervised learning technique featuring discriminative training that allows the speaker encoder to discern the differences in timbre among various voices. The triplet loss training necessitates a unique data sampling strategy.

During the training stage, we sample three utterance segments of equal length from the dataset: an anchor sample $x_{anc}$, a positive sample $x_{pos}$, and a negative sample $x_{neg}$. As illustrated in Figure 2, the anchor sample and the positive sample share the same timbre, while the negative sample is from a different speaker than the anchor. Therefore, let $nm$ represent the L2 normalization. We can use the speaker encoder outputs of the three samples to calculate a triplet loss:

$$e_{anc}, e_{pos}, e_{neg} = E_s(x_{anc}), E_s(x_{pos}), E_s(x_{neg}) \tag{12}$$

$$\begin{aligned} L_{tri} = E[nm(e_{anc}) * nm(e_{pos})^t] - \\ E[nm(e_{anc}) * nm(e_{neg})^t] + \delta \end{aligned} \tag{13}$$

The $\delta$ is a hyperparameter to control the speaker similarity. Denoting the VC model as $G_{vc}$, the total model's output can be described as follows:

$$y_1 = G_{vc}(x_{anc}, x_{neg}) \tag{14}$$

$$y_2 = G_{vc}(x_{anc}, x_{pos}) \tag{15}$$

### *F. Training Objective*

The training objective of the Pureformer-VC model includes VAE loss, AAM-softmax loss, and triplet loss. The total VAE loss is based on two outputs $y_1, y_2$ as shown in Figure 2 and can be denoted as:

$$L_{t-vae} = \lambda_1(L_{vae}(x_{anc}, y_1) + \lambda_2 L_{vae}(x_{anc}, y_2)) \tag{16}$$

The AAM-softmax loss can be computed using the three true labels of samples $C$ and the predictions of the speaker encoder:

$$L_{t-aam} = \sum_{c_i \in C} L_{aam}(c_i, x_i) \tag{17}$$

Finally, the triplet loss helps the speaker encoder to distinguish the embeddings. The total training objective is as follows:

$$L_{total} = L_{t-vae} + \lambda_3 L_{t-aam} + \lambda_4 L_{tri} \tag{18}$$

### *G. Vocoder*

The vocoder has the same structure as the HiFi-GAN generator. In our study, it was pre-trained on the same dataset used for the VC training.

## IV. Experiments and Results

### *A. Experimental Setup*

**Datasets and Feature Setup.** To evaluate the effectiveness of Pureformer-VC, we conducted a comparative experiment and an ablation study on VCTK [37] and AISHELL-3 [38] datasets. The VCTK corpus includes 109 English speakers, each reading about 400 utterances. The AISHELL-3 corpus contains roughly 85 hours of emotion-neutral recordings spoken by 218 native Chinese Mandarin speakers and 88035 utterances.

The mel-spectrogram extraction process must align with the pre-trained vocoder and adhere to the algorithm outlined in the HiFi-GAN framework [34]. The signal hyperparameters are defined as follows: 80 Mel frequency filters, 1024 FFT bins, a window length of 1024, a hop length of 256, a sampling rate of 22,050 Hz, and a maximum frequency of 8 kHz. The final input features are log-mel spectrograms, obtained by applying the logarithm to the extracted mel-spectrograms.

Considering batch sampling during the training stage, we randomly selected an utterance from one speaker. We then sampled two utterances from another speaker to create a training sample: $\{x_{anc}, x_{pos}, x_{neg}\}$. Five speakers are randomly chosen as unseen speakers for each corpus.

**Training Setup.** In the training stage, the batch size is 16. The learning rate is constant at $2 \times 10^{-4}$. The Pureformer-VC is trained by Adam optimizer [40] with $\beta_1 = 0.9, \beta_2 = 0.99, \epsilon = 1 \times 10^{-6}$. The $\lambda_1$ is set to 10 and $\lambda_2$ ranges from $1 \times 10 - 4$ to 1. Both the $\lambda_3, \lambda_4$ are set to 1. The $\delta$ is 0.3.

**Baseline Setup.** We compared Pureformer-VC with recent VC frameworks, such as AdaIN-VC [19], AutoVC [20], VQMIVC [39], MAIN-VC [30], RVC[1], and GPT-SoVITS[2]. The experiments are conducted in a many-to-many, one-shot (any-to-any) setup. We further evaluate the performance of the proposed model in cross-lingual VC, where the source utterance's language differs from that of the target language.

### *B. Metrics and Evaluation*

We assess the naturalness and intelligibility of the generated speech using subjective metrics, such as the Mean Opinion Score (MOS). Additionally, we utilize objective metrics to evaluate timbre similarity with the target speech, including the Voice Similarity Score (VSS) and Mel-Cepstral Distortion (MCD). Higher scores indicate greater effectiveness of the voice conversion (VC) system.

**Mean Opinion Score (MOS)**. The Mean Opinion Score (MOS) is a widely used metric for assessing the subjective quality of speech or audio. It is based on ratings from human listeners, who are asked to evaluate the quality of speech samples using a scale that typically ranges from 1 to 5. A higher MOS signifies better reconstruction quality.

**Voice similarity score (VSS)**. The Voice Similarity Score (VSS) is an objective metric that quantifies the degree of resemblance between generated speech and authentic or target speech in terms of timbre, tone, and voice quality. VSS is calculated based on embedding similarity derived from a pre-trained speaker verification model (Resemblyzer) [41]. Higher scores represent greater similarity, indicating improved voice conversion (VC) performance.

**Mel-cepstral distortion (MCD)**. MCD is an objective quantitative measure that evaluates the Mel-cepstral divergence between the source and generated utterances. The lower the MCD, the better the reconstruction effect.

TABLE I

Comparison of baseline and proposed methods for many-to-many and one-shot VC on the VCTK dataset (with a 95% confidence interval).

| 2*VC method | Many-to-many |  |  | One-shot |  |  |
|---|---|---|---|---|---|---|
|  | MCD↓ | MOS ↑ | VSS↑ | MCD↓ | MOS↑ | VSS↑ |
| AdaIN-VC [19] | 7.64 ± 0.24 | 3.11 ± 0.13 | 2.82 ± 0.19 | 7.38 ± 0.14 | 3.04 ± 0.21 | 2.45 ± 0.16 |
| AutoVC [20] | 6.68 ± 0.21 | 2.76 ± 0.16 | 2.45 ± 0.23 | 8.08 ± 0.15 | 2.68 ± 0.17 | 2.39 ± 0.14 |
| VQMIVC [39] | 6.24 ± 0.13 | 3.20 ± 0.14 | 3.32 ± 0.12 | 5.59 ± 0.10 | 3.16 ± 0.18 | 3.02 ± 0.18 |
| MAIN-VC [30] | 5.28 ± 0.11 | 3.44 ± 0.12 | 3.25 ± 0.16 | 5.42 ± 0.13 | 3.24 ± 0.18 | 3.29 ± 0.11 |
| RVC[1] | 4.75 ± 0.09 | 3.70 ± 0.07 | 3.72 ± 0.10 | 4.70 ± 0.12 | 3.41 ± 0.08 | 3.47 ± 0.09 |
| GPT-SoVITS[2] | 4.70 ± 0.07 | 3.70 ± 0.08 | 3.68 ± 0.10 | 4.68 ± 0.12 | 3.43 ± 0.09 | 3.50 ± 0.10 |
| Pureformer-VC(w/o AAM-softmax) | 5.25 ± 0.11 | 3.42 ± 0.12 | 3.05 ± 0.16 | 5.21 ± 0.12 | 3.20 ± 0.11 | 3.10 ± 0.11 |
| Pureformer-VC(w/o triplet) | 5.06 ± 0.09 | 3.25 ± 0.12 | 3.55 ± 0.14 | 4.92 ± 0.11 | 3.32 ± 0.15 | 3.21 ± 0.10 |
| Pureformer-VC | 4.95 ± 0.12 | 3.64 ± 0.13 | 3.56 ± 0.13 | 4.89 ± 0.10 | 3.36 ± 0.15 | 3.35 ± 0.12 |

TABLE II

Comparison of baseline and proposed methods for many-to-many and one-shot VC on the AISHELL-3 dataset (with a 95% confidence interval).

| 2*VC method | Many-to-many |  |  | One-shot |  |  |
|---|---|---|---|---|---|---|
|  | MCD↓ | MOS↑ | VSS↑ | MCD↓ | MOS↑ | VSS↑ |
| AdaIN-VC [19] | 7.29 ± 0.34 | 3.23 ± 0.21 | 2.76 ± 0.15 | 6.95 ± 0.20 | 3.13 ± 0.17 | 2.34 ± 0.11 |
| AutoVC [20] | 6.26 ± 0.25 | 2.85 ± 0.15 | 2.40 ± 0.19 | 6.64 ± 0.23 | 3.21 ± 0.20 | 2.36 ± 0.15 |
| VQMIVC [39] | 6.07 ± 0.16 | 3.18 ± 0.17 | 3.34 ± 0.24 | 5.35 ± 0.16 | 3.02 ± 0.24 | 3.13 ± 0.11 |
| MAIN-VC [30] | 5.06 ± 0.14 | 3.56 ± 0.16 | 3.29 ± 0.09 | 5.26 ± 0.11 | 3.35 ± 0.16 | 3.32 ± 0.17 |
| RVC[1] | 4.36 ± 0.12 | 3.74 ± 0.11 | 3.65 ± 0.12 | 4.34 ± 0.08 | 3.65 ± 0.11 | 3.60 ± 0.13 |
| GPT-SoVITS[2] | 4.28 ± 0.09 | 3.75 ± 0.14 | 3.69 ± 0.15 | 4.30 ± 0.13 | 3.68 ± 0.12 | 3.63 ± 0.12 |
| Pureformer-VC(w/o AAM-softmax) | 4.97 ± 0.16 | 3.58 ± 0.14 | 3.27 ± 0.12 | 5.05 ± 0.13 | 3.37 ± 0.12 | 3.28 ± 0.16 |
| Pureformer-VC(w/o triplet) | 4.66 ± 0.20 | 3.65 ± 0.15 | 3.32 ± 0.17 | 4.94 ± 0.22 | 3.45 ± 0.15 | 3.39 ± 0.24 |
| Pureformer-VC | 4.50 ± 0.15 | 3.69 ± 0.20 | 3.56 ± 0.19 | 4.64 ± 0.14 | 3.59 ± 0.16 | 3.50 ± 0.18 |

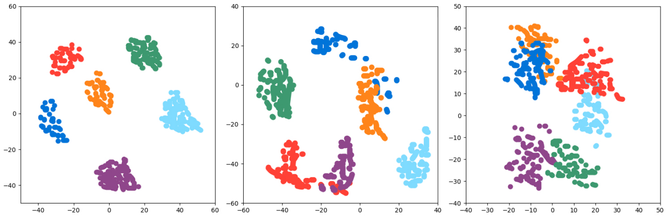
During the testing phase, both many-to-many and one-shot (any-to-any) VC are conducted using non-parallel data. For evaluation, 10 source/target speech pairs are fed into each VC model under two scenarios. After this, 5 participants are invited to rate the speech samples.

### C. Experimental Results

Table I and II present a comparative analysis of the Pureformer-VC against baseline methods in both many-to-many and one-shot VC settings across both datasets.

**MCD and MOS metrics.** The MOS and MCD can evaluate the effectiveness of speech reconstruction, collectively referred to as reconstruction metrics. Since the original sampling rate of the AISHELL-3 dataset is higher than that of the VCTK dataset, the quality of speech reconstruction is superior, resulting in better reconstruction metrics for the AISHELL-3 dataset. The Pureformer-VC outperforms the encoder-decoder baseline models in both many-to-many and one-shot experimental configurations with respect to reconstruction metrics. However, when compared to state-of-the-art methods like RVC and GPT-SoVITS, the Pureformer-VC still exhibits a slight performance gap. These results showcase the effectiveness of the pure transformer architecture.

**VSS metric.** The VSS evaluation measures the similarity between the generated target timbre and the actual target timbre using a resemblyzer. Compared to the four classic encoder-decoder-based VC baselines, the Pureformer-VC surpasses them in the VSS metric. However, a slight gap remains in the VSS performance between PVC and the state-of-the-art models RVC and GPT-SoVITS. As shown in the results, we further investigated the impact of removing either the AAM-Softmax loss or the triplet loss from the training objectives to assess the model's ability to represent timbre in the embedded vectors during training. It is evident that without these two losses, the proposed model's VSS stays relatively consistent with that of the baseline model. Therefore, incorporating these losses helps enhance the model's VC expressiveness.



(a) Pureformer-VC (b) w/o triplet loss w/o AAM-softmax

Fig. 3. The visualization of speaker representations extracted from 6 unseen speakers' utterances.

### D. Ablation Study

We conduct ablation experiments to validate the effects of triplet loss and AAM-softmax loss on disentanglement. We set up the following models: (a) the Pureformer-VC model, (b) the Pureformer-VC model without triplet loss (w/o triplet), and (c) the Pureformer-VC model without AAM-softmax loss (w/o AAM-softmax). We used the resemblyzer to detect synthetic speech and evaluate conversion quality. It assigns detection scores to fake (i.e., the VC model's experimental outputs) and authentic utterances from the target speaker after learning the target's characteristics from ten additional genuine utterances. A higher score signifies a closer resemblance in timbre and superior speech quality. The results are detailed in Table III. We found that the decision scores of our method are higher than those of the best baseline model, MAIN-VC. Furthermore, the experiments demonstrate that both triplet loss and AAM-Softmax contribute to improving the accuracy of timbre generation.

For a visual evaluation of each model's disentanglement capability, the t-SNE scatter plots of the speaker representations are shown in Figure 3. The AAM-softmax loss has a significant impact on the clustering of speaker embedding vectors, while the triplet loss helps create more distinct boundaries between categories.

TABLE III
FAKE DETECTION SCORE COMPARISON FOR ABLATION STUDY ABLATION

| Method | Detection Score↑ | |
|---|---|---|
| | VCTK | AISHELL-3 |
| Pureformer-VC | 0.75 ± 0.08 | 0.73 ± 0.08 |
| w/o triplet loss | 0.65 ± 0.12 | 0.68 ± 0.12 |
| w/o AAM-Softmax | 0.52 ± 0.12 | 0.56 ± 0.12 |
| MAIN-VC | 0.74 ± 0.01 | 0.73 ± 0.01 |
| RVC | 0.78 ± 0.01 | 0.75 ± 0.01 |
| GPT-SoVITS | 0.80 ± 0.01 | 0.77 ± 0.01 |

### *E. Cross-lingual VC*

Pureformer VC is also capable of performing cross-lingual voice conversion (CVC), where the source and target utterances are in different languages. We trained Pureformer-VC on a mixture of the AISHELL-3 and VCTK datasets. Due to varying pronunciation habits, the reconstruction metrics ($MOS = 3.15, MCD = 5.12$) and VSS (2.95) scores for cross-lingual voice conversion are lower than those for monolingual VC experiments. The bilingual timbre experiments suggest that additional latent variables may be necessary to decouple the languages using special encoders for improved conversion performance.

## V. CONCLUSION

In this paper, we present a novel approach to voice conversion (VC) by leveraging a pure transformer network designed as a VAE encoder-decoder framework called Pureformer-VC. Within the decoder, we integrate a styleformer module, which enhances the model's capacity for style transfer. Additionally, we improve the effectiveness of the speaker encoder by incorporating triplet loss and AAMSoftmax loss. These enhancements significantly boost the model's ability to capture and represent the nuances of different speaking voices, resulting in more accurate and robust VC. In conclusion, the Pureformer-VC model, fortified by the strategic application of specialized loss functions and style adaptation mechanisms, represents a significant advancement in the field of VC.

## REFERENCES

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

[2] E. Helander, T. Virtanen, J. Nurminen, and M. Gabbouj, "Voice conversion using partial least squares regression," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 5, pp. 912–921, 2010.

[3] 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.

[4] D. Erro, A. Moreno, and A. Bonafonte, "Voice conversion based on weighted frequency warping," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 5, pp. 922–931, 2009.

[5] R. Aihara, R. Ueda, T. Takiguchi, and Y. Ariki, "Exemplar-based emotional voice conversion using non-negative matrix factorization," in *Signal and Information Processing Association Annual Summit and Conference (APSIPA), 2014 Asia-Pacific*. IEEE, 2014, pp. 1–7.

[6] T. Kaneko, H. Kameoka, K. Tanaka, and N. Hojo, "Cyclegan-vc3: Examining and improving cyclegan-vcs for mel-spectrogram conversion," *arXiv preprint arXiv:2010.11672*, 2020.

[7] ——, "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.

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

[9] T. Kaneko, H. Kameoka, K. Tanaka, and N. Hojo, "Stargan-vc2: Rethinking conditional methods for stargan-based voice conversion," *arXiv preprint arXiv:1907.12279*, 2019.

[10] Y. A. Li, A. Zare, and N. Mesgarani, "Starganv2-vc: A diverse, unsupervised, non-parallel framework for natural-sounding voice conversion," *arXiv preprint arXiv:2107.10394*, 2021.

[11] M. Chen, Y. Shi, and T. Hain, "Towards low-resource stargan voice conversion using weight adaptive instance normalization," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 5949–5953.

[12] 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.

[13] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, "Analyzing and improving the image quality of stylegan," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 8110–8119.

[14] B. Van Niekerk, M.-A. Carbonneau, J. Zaïdi, M. Baas, H. Seuté, and H. Kamper, "A comparison of discrete and soft speech units for improved voice conversion," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 6562–6566.

[15] E. Casanova, J. Weber, C. D. Shulby, A. C. Junior, E. Gölge, and M. A. Ponti, "Yourtts: Towards zero-shot multi-speaker tts and zero-shot voice conversion for everyone," in *International conference on machine learning*. PMLR, 2022, pp. 2709–2720.

[16] M. Baas, B. van Niekerk, and H. Kamper, "Voice conversion with just nearest neighbors," *arXiv preprint arXiv:2305.18975*, 2023.

[17] 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.

[18] C. H. Chan, K. Qian, Y. Zhang, and M. Hasegawa-Johnson, "Speechsplit2. 0: Unsupervised speech disentanglement for voice conversion without tuning autoencoder bottlenecks," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 6332–6336.

[19] 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.

[20] K. Qian, Y. Zhang, S. Chang, X. Yang, and M. Hasegawa-Johnson, "Autovc: Zero-shot voice style transfer with only autoencoder loss," in

*International Conference on Machine Learning*. PMLR, 2019, pp. 5210–5219.

[21] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu *et al.*, "Conformer: Convolution-augmented transformer for speech recognition," *arXiv preprint arXiv:2005.08100*, 2020.

[22] Z. Gao, S. Zhang, I. McLoughlin, and Z. Yan, "Paraformer: Fast and accurate parallel transformer for non-autoregressive end-to-end speech recognition," *arXiv preprint arXiv:2206.08317*, 2022.

[23] Z. Yao, L. Guo, X. Yang, W. Kang, F. Kuang, Y. Yang, Z. Jin, L. Lin, and D. Povey, "Zipformer: A faster and better encoder for automatic speech recognition," *arXiv preprint arXiv:2310.11230*, 2023.

[24] K. An, Z. Li, Z. Gao, and S. Zhang, "Paraformer-v2: An improved non-autoregressive transformer for noise-robust speech recognition," *arXiv preprint arXiv:2409.17746*, 2024.

[25] Y. Zhang, Z. Lv, H. Wu, S. Zhang, P. Hu, Z. Wu, H.-y. Lee, and H. Meng, "Mfa-conformer: Multi-scale feature aggregation conformer for automatic speaker verification," *arXiv preprint arXiv:2203.15249*, 2022.

[26] T. Liu, R. K. Das, K. A. Lee, and H. Li, "Mfa: Tdnn with multi-scale frequency-channel attention for text-independent speaker verification with short utterances," in *ICASSP 2022-2022 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2022, pp. 7517–7521.

[27] E. Kim and H. Seo, "Se-conformer: Time-domain speech enhancement using conformer." in *Interspeech*, 2021, pp. 2736–2740.

[28] S. Abdulatif, R. Cao, and B. Yang, "Cmgan: Conformer-based metric-gan for monaural speech enhancement," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2024.

[29] A. Hermans, L. Beyer, and B. Leibe, "In defense of the triplet loss for person re-identification," *arXiv preprint arXiv:1703.07737*, 2017.

[30] P. Li, J. Wang, X. Zhang, Y. Zhang, J. Xiao, and N. Cheng, "Main-vc: Lightweight speech representation disentanglement for one-shot voice conversion," *arXiv preprint arXiv:2405.00930*, 2024.

[31] K. Shao, K. Chen, M. Baas, and S. Dubnov, "knn-svc: Robust zero-shot singing voice conversion with additive synthesis and concatenation smoothness optimization," in *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2025, pp. 1–5.

[32] D. Ke, W. Yao, R. Hu, L. Huang, Q. Luo, and W. Shu, "A new spoken language teaching tech: Combining multi-attention and adain for one-shot cross language voice conversion," in *2022 13th International Symposium on Chinese Spoken Language Processing (ISCSLP)*. IEEE, 2022, pp. 101–104.

[33] X. Wu, Z. Hu, L. Sheng, and D. Xu, "Styleformer: Real-time arbitrary style transfer via parametric style composition," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 14 618–14 627.

[34] J. Kong, J. Kim, and J. Bae, "Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis," *Advances in neural information processing systems*, vol. 33, pp. 17 022–17 033, 2020.

[35] Q. Zhu, J. Su, W. Bi, X. Liu, X. Ma, X. Li, and D. Wu, "A batch normalized inference network keeps the kl vanishing away," *arXiv preprint arXiv:2004.12585*, 2020.

[36] T. Salimans and D. P. Kingma, "Weight normalization: A simple reparameterization to accelerate training of deep neural networks," *Advances in neural information processing systems*, vol. 29, 2016.

[37] C. Veaux, J. Yamagishi, and S. King, "Vctk corpus: English multi-speaker corpus for cstr voice cloning," *arXiv preprint arXiv:2012.11929 [cs.Speech]*, 2020.

[38] Y. Shi, H. Bu, X. Xu, S. Zhang, and M. Li, "Aishell-3: A multi-speaker mandarin tts corpus and the baselines," *arXiv preprint arXiv:2010.11567*, 2020.

[39] 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.

[40] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[41] B. Desplanques, J. Thienpondt, and K. Demuynck, "Ecapa-tdnn: Emphasized channel attention, propagation and aggregation in tdnn based speaker verification," *arXiv preprint arXiv:2005.07143*, 2020.