Audio Engineering Society

Conference Paper 23

Presented at the AES International Conference on Machine Learning and Artificial Intelligence for Audio
2025 September 8–10, London, UK

*This paper was peer-reviewed as a complete manuscript for presentation at this conference. This paper is available in the AES E-Library (http://www.aes.org/e-lib), all rights reserved. Reproduction of this paper, or any portion thereof, is not permitted without direct permission from the Journal of the Audio Engineering Society.*

# Improved Singing Voice Conversion with Frame-Level Content and Melody-Informed Speaker Embeddings Using Cross-Attention

Jih-Wei Yeh[1], Elaine M. Liu[1], and Yi-Wen Liu[1]

[1]*Department of Electrical Engineering, National Tsing Hua University, Taiwan*

Correspondence should be addressed to Jih-Wei Yeh (wilsun890126@gmail.com)

**ABSTRACT**

Singing voice conversion (SVC) converts audio sung by a source voice into the timbre of a target voice without changing the linguistic content or melody of the sung utterance. The popular SVC paradigm is to extract content features from the source and condition on timbre features extracted from the target to generate converted audio. In recent years, great strides have been made in SVC with advancements in content representations. However, what constitutes an accurate representation of timbre is still an emerging topic. Currently, most SVC systems use a single speaker embedding to represent one singer, but this disregards the large variations in timbre characteristics even for the same person when singing in different techniques and vocal ranges. In this paper, we propose a cross-attention mechanism to generate frame-wise speaker timbre embeddings informed by linguistic content and melody; we aim to generate timbre features that more accurately reflect how the voice changes with different vocal ranges and enunciations at different times. Objective metrics and subjective listening tests of our converted singing audio show improvements in enunciation clarity and voice similarity, while expressiveness is a more elusive challenge.

## 1 Introduction

Singing voice conversion (SVC) involves transforming source audio that includes phonetic content, rhythm, melody, and singing techniques into output audio that retains the content while adopting the timbre of a target singer. Recent studies focus on non-parallel singing voice conversion, where a sung utterance audio is reconstructed during training, and the conversion is inferred by prompting a singer ID or embedding vector. SVC has also benefited from advancements in the extraction of content-related and timbre-related acoustic features.

In the realm of content-related acoustic features, numerous pre-trained models developed for automatic speech recognition (ASR), such as phonetic posteriorgrams (PPGs) [1], Wav2Vec2.0 [2], HuBERT [3], and ContentVec [4], have proven effective in capturing phonetic content from an utterance. Another important content feature, particularly for singing voice conversion, is the song melody. Established algorithms, such as DIO in the WORLD vocoder [5], as well as deep-learning-based algorithms such as CREPE [6] and RMVPE [7], are effective at extracting the fundamental frequency $f_0$ contour from singing audio. These content-related acoustic features can be generated frame-wise by employing a sliding window method, similar to the short-

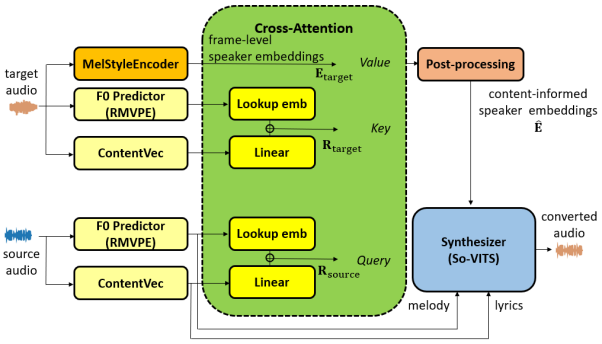



**Fig. 1:** Our proposed singing voice conversion framework using frame-wise speaker embeddings informed by ContentVec and $f_0$.

time Fourier transform, ensuring alignment across different features.

As for the timbre-related acoustic features, early work [8] represented a singer using a one-hot-like identity vector, but this approach has limitations on feature reusability and generalizability. First, one-hot-type vectors simply prompt the system to toggle between a fixed set of voices seen in training (any-to-many voice conversion). Second, such ID representations do not capture timbre information in a latent space where singers with similar voices tend to cluster with small distances. Such a latent space would allow for creativity with interpolation for voice morphing [9], or extrapolation for an unseen target voice (any-to-any voice conversion) [10].

Recent research has addressed this by replacing one-hot-type vectors with meaningful speaker embeddings such as ECAPA-TDNN speaker embeddings [11] or x-vectors [12] which are pre-trained for speaker recognition tasks in ASR. In our work, the term "speaker embedding" is borrowed from speech literature to refer to singer timbre representation. With the x-vector system, only one static vector is used to represent the timbre of an entire audio clip regardless of length, but this fails to capture timbre variability over time and results in a lack of adaptability to different enunciations and frequency ranges when singing.

In this paper, we propose a method to generate speaker embeddings from frame-level timbre information, in congruence with the frame-level approach for content features. Our approach shown in Figure 1 applies cross-attention to the lyrics and melody of the utterance and predicts the corresponding speaker embedding for each frame, thus enabling better adaptability to timbre variations over time. Additionally, we train our model on a more timbre-diverse dataset with 100 Mandarin singers, enabling better timbre generalization across a wide range of voices. Consequently, our model surpasses previous approaches in terms of voice similarity, vocal expressiveness, and enunciation clarity.

Inspired by the self-supervised learning paradigm, our training strategy resembles masked autoencoder frameworks such as HuBERT [3], in which parts of the input are intentionally masked and reconstructed to encourage the model to learn robust internal representations. Specifically, we mask the current and surrounding frames during training and learn to reconstruct the current frame's timbre embedding from other non-masked, content-related frames. This enables the model to discover meaningful frame-wise timbre information without relying on external labels.

## 2 Related work

### 2.1 So-VITS-SVC

So-VITS[1] is a conditional variational autoencoder (VAE) with a flow-based module that achieves state-of-the-art naturalness and timbral similarity in synthesizing singing voices of target singers [13]. Its core architecture includes one decoder and two encoders: a prior encoder that maps acoustic features into a latent space, and a posterior encoder that encodes mel-spectrograms of target utterances. A Kullback-Leibler (KL) divergence is applied to the two latent distributions to align them during training. The decoder reconstructs the singing voice from the latent representation, effectively serving as the generator within the VAE architecture.

To further enhance generation quality, So-VITS incorporates a generative adversarial network (GAN) component during training. Specifically, a multi-period discriminator is adopted, similar to that used in HiFi-GAN [14], to distinguish real from synthesized waveforms. This introduces an adversarial loss that guides the decoder to produce more natural-sounding audio.

Beyond So-VITS' architectural strengths, its success is also attributed to the use of ContentVec as a robust content feature extractor.

[1] https://github.com/svc-develop-team/so-vits-svc.git

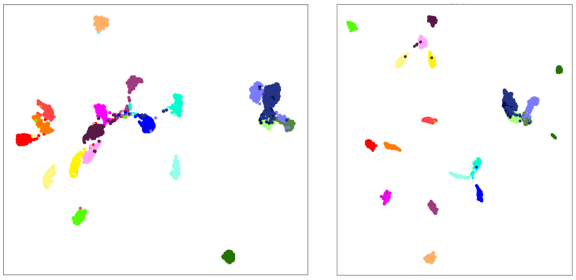
### 2.2 ContentVec

ContentVec [4], adapted from HuBERT, excels at preserving content information while removing speaker-related details. It is comprised of three components: a predictor and two speech representation networks, referred to as the *Student* and *Teacher*. *Teacher* uses a pre-trained, unsupervised model that generates speech representations, which are then clustered using K-means. These clusters serve as self-supervised learning labels for *Student*. *Student*, another speech representation network, is trained to focus on learning content-related information. The predictor classifies the categories of the speech representations generated by the *Student*. Both the *Student* and the predictor are trained to minimize cross-entropy loss between the clustered labels and the predicted labels. ContentVec outperforms other models with similar functionality by applying timbre transformations to both *Student* and *Teacher* networks, which allows *Student* and predictor to focus more effectively on content, thereby preserving more content information while disentangling speaker information.

### 2.3 Mel-Style Encoder

We choose to use the mel-style encoder proposed by Min et al [15] in place of the x-vector to generate speaker embeddings for timbre conditioning in So-VITS. This decision is based on our observation that the mel-style encoder demonstrates better voice distinction on the M4Singer [16] dataset, which consists of 20 singers across four voice types: soprano, alto, tenor, and bass. It produces tighter clustering compared to the x-vector in 2D visualizations, where the n-dimensional speaker embeddings are reduced using UMAP [17], as shown in Figure 2. In the visualization, the 20 singers are clustered in different colors, clearly reflecting inter-singer separability. Details of these embeddings are described as follows.

The mel-style encoder takes a mel-spectrogram as input and generates a vector that captures the speaker's timbre characteristics for a given utterance. This encoder consists of four sequential modules: spectral processing, temporal processing, multi-head self-attention, and a time-averaging module. The spectral and temporal processing modules transform spectral information from the frequency domain into a latent space, while capturing temporal patterns in the time domain. Following this, multi-head self-attention is applied to highlight global patterns across the entire utterance. Finally, the time-averaging module produces a speaker embedding of the entire utterance to represent the timbre information of the speaker. We adopt the first three modules of this encoder: spectral processing, temporal processing, and multi-head self-attention, while discarding the fourth time-averaging module, in order to generate frame-level speaker embeddings.



**(a)** X-vector **(b)** Mel-Style Embeds

**Fig. 2:** 2D Visualization of Speaker Embeddings.

### 2.4 X-Vector

The x-vector [12] is a powerful module pre-trained on thousands of speakers from the VoxCeleb [18] dataset. X-vectors have been well-established in forensics literature [19] as effective speaker representations with good accuracy in distinguishing different voices. The x-vector encoder takes audio as input and outputs a 512-dimensional speaker embedding. The encoder module was pre-trained on VoxCeleb for the purpose of automatic speaker recognition, and the output of the second-to-last hidden layer is used as a timbre feature vector, since the network has learned to distill information relevant to distinguishing one voice from another.

## 3 Methods

The overall architecture of our model for generating frame-wise speaker embeddings is shown in Figure 3. The model consists of a mel-style encoder, a multi-head cross-attention module informed by ContentVec and the $f_0$ contour, and a post-processing module. In this section, we provide a detailed explanation of each component.

### 3.1 Frame-wise Acoustic Feature Preparation

Consider a singing utterance $\boldsymbol{x} \in \mathbb{R}^{1\times L_a}$, where $L_a$ represents the samples in the audio waveform. We processed

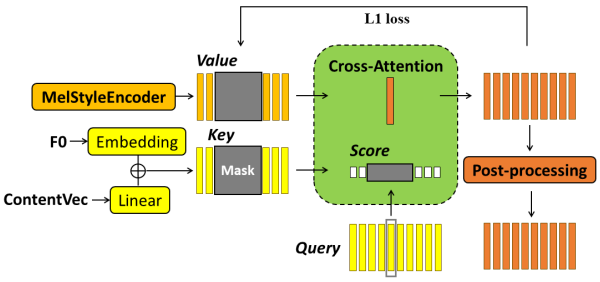



**Fig. 3:** Multi-head cross-attention on ContentVec and $f_0$.

audio clips of duration 5 to 10 seconds with a sampling rate of 44.1kHz, Hann window, frame length of 2048 samples, and hop size of 512.

Three crucial frame-wise acoustic features are generated from the utterance by the following modules: 1) the timbre-related Mel-Style Encoder $e(\cdot) : \mathbb{R}^{1\times L_a} \to \mathbb{R}^{C_e\times L_T}$, 2) the fundamental frequency predictor $f(\cdot) : \mathbb{R}^{1\times L_a} \to \mathbb{R}^{2\times L_T}$, and 3) the lyrics representation content-related encoder $c(\cdot) : \mathbb{R}^{1\times L_a} \to \mathbb{R}^{C_c\times L_T}$, where $L_T$ represents the number of frames, and $C_e$=512, $C_c$=768 represent the embedding dimensions, respectively.

In order to generate frame-wise speaker embeddings for timbre information, we combine the mel-spectrogram and our revised mel-style encoder (where we kept the first three modules but removed the time-averaging module) as our timbre-related encoder. The output is denoted as:

$$\mathbf{E} = e(\boldsymbol{x}) \in \mathbb{R}^{C_e\times L_T} \tag{1}$$

Additionally, we utilize two pre-trained models: ContentVec and RMVPE, to extract content-related features. The ContentVec module leverages self-supervised learning techniques and demonstrates robust performance in downstream tasks such as ASR and voice conversion [4], and is used here to represent lyrics information:

$$\mathbf{C} = c(\boldsymbol{x}) \in \mathbb{R}^{C_c\times L_T} \tag{2}$$

We use the RMVPE module to represent melody, as it excels at generating reliable $f_0$ contours, outperforming traditional algorithms and earlier deep-learning-based models [7]. The output is denoted as:

$$\mathbf{F} = f(\boldsymbol{x}) = (\boldsymbol{f_0}, \boldsymbol{v}) \tag{3}$$

where $\boldsymbol{f_0} \in \mathbb{R}^{L_T}$ is the fundamental frequency contour and $\boldsymbol{v} \in \{0,1\}^{L_T}$ contains binary indicators of voiced or unvoiced in each frame.

### 3.2 Multi-head Cross-Attention

The multi-head cross-attention [20] module shown in Figure 3, aims to generate frame-wise speaker embeddings for timbre representations, informed by ContentVec (lyrics and enunciation information) and the $f_0$ contour (melody information).

The general equation for cross-attention is given by

$$\text{CrossAttention}(\mathbf{Q},\mathbf{K},\mathbf{V}) = \mathbf{V}\cdot\text{softmax}\left(\frac{\mathbf{K}^\top\mathbf{Q}}{\sqrt{d_k}}\right) \tag{4}$$

where queries are column vectors in $\mathbf{Q} \in \mathbb{R}^{d_k\times n_q}$, and similarity scores are calculated with keys that are column vectors in $\mathbf{K} \in \mathbb{R}^{d_k\times n_k}$. Values are column vectors in $\mathbf{V} \in \mathbb{R}^{d_v\times n_k}$, resulting in a $\mathbb{R}^{d_v\times n_q}$ matrix, where each column is a weighted sum of value column vectors corresponding to its query vector .

Figure 3 shows sets of frame-level vectors. In this process, the *query* $\mathbf{Q}$ and *key* $\mathbf{K}$ in the multi-head cross-attention module are acoustic features related to content, obtained by projecting ContentVec $\mathbf{C}$ and the pitch information $\mathbf{F}$ into a $C_R$=192-dimensional latent space and adding:

$$\mathbf{R} = \mathscr{B}_1(\boldsymbol{f_0}) + \mathscr{B}_2(\boldsymbol{v}) + \text{proj}(\mathbf{C}) \in \mathbb{R}^{C_R\times L_T} \tag{5}$$

where $\mathscr{B}_1$ and $\mathscr{B}_2$ are lookup tables that transform the 256 discrete frequency bins in $\boldsymbol{f_0}$ and the 2 categories of voiced/unvoiced in $\boldsymbol{v}$ into $C_R$-dimensional vectors, respectively.

We predict the $i$th frame's speaker embedding as follows:

$$\hat{\mathbf{E}}_i = \text{CrossAttention}(\mathbf{R}_i, \mathbf{R}_{\mathscr{T}\setminus\mathscr{M}_i}, \mathbf{E}_{\mathscr{T}\setminus\mathscr{M}_i}) \in \mathbb{R}^{C_e\times 1} \tag{6}$$

where $\mathbf{R}_i \in \mathbb{R}^{C_R\times 1}$ is the $i$th column of $\mathbf{R}$, representing the lyrics and melody content of the current $i$th frame. The set of all frame indices is denoted as $\mathscr{T} = \{1,\ldots,L_T\}$, and $\mathscr{M}_i = \{i-k,\ldots,i+k\}$ defines a symmetric mask of approximately 0.5 seconds centered on the $i$th frame. The matrices $\mathbf{R}_{\mathscr{T}\setminus\mathscr{M}_i} \in \mathbb{R}^{C_R\times N}$ and $\mathbf{E}_{\mathscr{T}\setminus\mathscr{M}_i} \in \mathbb{R}^{C_e\times N}$ denote the masked-out content and speaker embedding sequences, respectively, where $N = L_T - (2k+1)$ in the general case. At the boundaries, the

mask is clipped to avoid invalid indices. Specifically, when $i = 1$, the mask becomes $\mathscr{M}_1 = \{1, \ldots, 1+k\}$; when $i = L_T$, we set $\mathscr{M}_{L_T} = \{L_T - k, \ldots, L_T\}$. In such cases, $N$ is adjusted accordingly, for instance, to $N = L_T - (k+1)$ when only one side of the mask is clipped.

The frame-wise *key-value* pairs consist of content-related acoustic features and speaker embeddings, respectively. Our concept is to utilize the current $i$th frame's acoustic content information $\mathbf{R}_i$ to query and find other frames that have similar content keys in $\mathbf{R}_{\mathscr{T} \setminus \mathscr{M}_i}$, in order to combine those frames' corresponding speaker embedding values in $\mathbf{E}_{\mathscr{T} \setminus \mathscr{M}_i}$, as a representation of the current frame's timbre.

During training, we encourage the cross-attention module to reconstruct the current frame's speaker embedding from the speaker embeddings of other frames with relevant content keys. To achieve this, both the *query* and the *key* are set to the same clip's content-related acoustic features. Close-proximity frames have near-identical content features, and we want our system to learn how to query for similar but *different* content. Therefore, we apply a mask $\mathscr{M}_i$ that spans approximately 0.5 seconds centered around the $i$th frame to ignore the current and neighboring frames' *key-value* pairs during training. For each frame's *query*, we obtain a relevance score with all the non-masked $\mathscr{T} \setminus \mathscr{M}_i$ frames' *keys*. High relevance means the current frame and another frame have similar content (word, melody) acoustic features, so intuitively these frames should have relevant speaker embeddings. These scores are then used as weights to predict the current frame's speaker embedding $\hat{\mathbf{E}}_i$ using a linear combination of the non-masked frames' speaker embeddings.

During training, we can apply the $L_1$ loss of $L_1(\hat{\mathbf{E}}_i, \mathbf{E}_i)$ between the predicted $\hat{\mathbf{E}}_i$ and original $\mathbf{E}_i$ frame-wise speaker embeddings, with the original serving as the ground truth.

During inference, no mask is needed since *query* and *key* originate from different audio, as shown in Figure 1. Here, the *query* is the content of the source audio, and the *key-value* pairs are the content and speaker embeddings from the target audio.

$$\hat{\mathbf{E}}_i = \text{CrossAttention}(\mathbf{R}_{\text{source},i}, \mathbf{R}_{\text{target}}, \mathbf{E}_{\text{target}}) \quad (7)$$

Thus, when the source content has changes in singing enunciations and pitch range, we can find relevant target timbre embeddings that reflect those enunciations and ranges.

### 3.3 Post-processing on Frame-wise Speaker Embeddings

Through the cross-attention module, we obtain frame-wise speaker embeddings informed by ContentVec and the $f_0$ contour, which can be directly used to condition the variational autoencoder in the So-VITS SVC model. However, before these embeddings are input into SoVits, we introduce a post-processing module. Since each frame is predicted independently by the cross-attention mechanism, the resulting frame-wise speaker embeddings are unaware of the embeddings of other frames, even those of neighboring ones, which may lead to perceptible discontinuities.

To address this, we apply 1D convolution to refine the frame-wise speaker embeddings generated by the cross-attention module. Each row in $\hat{\mathbf{E}} \in \mathbb{R}^{C_e \times L_T}$ is treated as a time series of length $L_T$, and a sliding kernel is applied so that neighboring values can be used to get a smoother result. In particular, our post-processing module contains one layer of 1D convolution, each layer with $C_e$ learnable sliding kernels of size 5, shift 1, and zero-padding. We denote our post-processing smoothing operation by $s(\cdot) : \mathbb{R}^{C_e \times L_T} \to \mathbb{R}^{C_e \times L_T}$.

Thus, we modify the loss function used in training to $L_1(s(\hat{\mathbf{E}}), \mathbf{E})$.

## 4 Experiments

In this section, we first describe the composition of the dataset and conduct an ablation study to compare our proposed model with the original So-VITS. Section 5 discusses our subjective and objective assessments.

**Table 1:** Details of the Datasets

| Dataset | Singers (F/M) | Duration | Sampling rate |
|---|---|---|---|
| **Mpop600** | 4 (2/2) | 12.3 hrs | 48 kHz |
| **M4Singer** | 20 (10/10) | 29.7 hrs | 48 kHz, 96 kHz |
| **OpenSinger** | 76 (48/28) | 51.9 hrs | 44.1 kHz |
| **Combined** | 100 (60/40) | 14.8 hrs | 44.1 kHz |

Audio used for training and inference was taken from three Mandarin singing corpora: our lab's internal MPop600 [21] dataset, M4Singer [16], and OpenSinger [22]. The combined dataset includes 100 singers, with 60 females and 40 males. Details about

the singers, sampling rate, and duration are provided in Table 1. We segmented all audio into 5- to 10-second clips.

The original So-VITS model relies on a one-hot-like speaker identity representation, which limits the model to learning only a few singers at a time and requires tens of minutes to a few hours of training audio per singer (any-to-many conversion). In our experiment, we replace their approach of few singers with many samples each, to our approach of many singers with few samples each: we use comparatively many singers (60 females, 40 males) with smaller training audio duration per singer (9 minutes each). This allows us to investigate how well the model can generalize across a diverse range of singer timbres, despite the limited duration of data per singer (toward any-to-any conversion).

Using the combined dataset, we trained the singing voice conversion model with three different types of speaker embeddings, as shown in the tables in section 5.

1. **So-VITS** The baseline model is the original So-VITS conditioned on x-vectors.
2. **MelStyle-SVC** The second model is So-VITS conditioned on clip-wise mel-style embeddings, which are obtained by taking the temporal average of frame-wise mel-style embeddings from the target timbre audio.
3. **CrossAtt-SVC** Our proposed model, however, is conditioned on frame-wise mel-style embeddings, which are informed by content-related acoustic features, using multi-head cross-attention and post-processed by a 1D convolution.

We trained each system in an end-to-end manner. All models were trained with the AdamW optimizer on two RTX-3090 GPUs, using a mini-batch size of 20 for 120,840 steps. The learning rate was set to $10^{-4}$.

## 5 Results and Discussion

In this section, we evaluate our results objectively and subjectively. Our objective metrics include Mel-Cepstral Distortion (MCD), Modulation Spectra Distance (MSD) [23], and cosine similarity of speaker embeddings (SpkCS) via x-vector and d-vector [2] [24, 9].

Because singing is highly subjective, we also rely on human perception in subjective listening tests and conduct mean opinion score (MOS) assessments on a 1-to-5 scale (5 being best) for the following three criteria:

- **Clarity of Articulation** refers to how clear the pronunciation of each word is in the audio file.
- **Timbre Similarity** measures how closely the timbre of the audio matches that of the real singer.
- **Vocal Expressiveness** assesses whether the singing demonstrates strength and control over the voice for more expressiveness throughout the performance.

We assess the systems in two parts. Part 1 focuses on singing voice reconstruction, where a sung utterance is used as both the source and target audio, and its own speaker embedding is employed to reconstruct the singing voice with the same content. Part 2 involves singing voice conversion, where the source singing audio is transformed into another timbre using a different target's speaker embedding. A demo page[3] is available.

**Table 2:** Part 1 Reconstructions Objective Metrics

| | MCD ↓ | MSD ↓ | SpkCS ↑ |
|---|---|---|---|
| So-VITS | 7.802 | 0.4945 | 0.9785/0.9233 |
| MelStyle-SVC | **7.614** | 0.4863 | 0.9885/0.9413 |
| CrossAtt-SVC | 7.755 | **0.4412** | **0.9909**/**0.9482** |

We begin by evaluating the upper-bound performance of each model in the part 1 reconstruction task. Objective metrics are shown in Table 2. We can observe improvements across all three objective metrics after replacing the x-vector speaker embedding with the mel-style embedding. Our proposed model shows progress in cosine similarity of speaker embeddings and particularly in Modulation Spectra Distance, with only a slight degradation in MCD. This suggests that the cross-attention mechanism enhances speaker similarity while allowing the reconstruction of more detailed varying spectro-temporal components.

In our subjective assessment, we collected data from 23 listeners via an online questionnaire. In the Part 1

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

[3] https://m105.nthu.edu.tw/~s105061807/wilsun/xattn.html

**Table 3:** Part 1 Reconstructions
Subjective MOS Scores

| | Clarity | Similarity | Expressiveness |
|---|---|---|---|
| Ground Truth | $4.48 \pm 0.28$ | $4.03 \pm 0.19$ | $4.14 \pm 0.18$ |
| So-VITS | $3.77 \pm 0.37$ | $3.74 \pm 0.31$ | $3.77 \pm 0.32$ |
| MelStyle-SVC | $4.01 \pm 0.26$ | $3.73 \pm 0.25$ | $3.73 \pm 0.27$ |
| CrossAtt-SVC | $\mathbf{4.17 \pm 0.27}$ | $\mathbf{3.80 \pm 0.16}$ | $\mathbf{3.81 \pm 0.30}$ |

reconstruction task, 10 audio examples were each reconstructed by our 3 systems, along with the ground truth audio sung by real singers. In the Part 2 voice conversion task, 10 audio examples were also converted into another timbre by our 3 systems. These 70 audio examples were MOS-evaluated by listeners for clarity, similarity, and expressiveness. The audio examples used in evaluation had been held out from the training set. During the blind listening tests, the real singers' audio files and the synthesized outputs from three models were randomly ordered.

Table 3 further demonstrates that our model achieves the best performance across all three subjective MOS metrics, especially in clarity of articulation.

Next, we apply objective and subjective metrics to our part 2 singing voice conversion task. The results for these evaluations are presented in Table 4. For objective metrics, we use the x- and d-vector cosine similarities, since MCD and MSD would require unavailable ground truths. Subjective MOS scores for clarity, similarity, and expressiveness are also shown. We see that our proposed model still outperforms in cosine similarity of speaker embeddings, clarity and similarity with a slight drop in expressiveness.

**Table 4:** Part 2 Voice Conversions
Objective SpkCS and Subjective MOS

| | SpkCS | Clarity | Similarity | Expressiveness |
|---|---|---|---|---|
| So-VITS | 0.9615/0.8491 | $3.89 \pm 0.24$ | $3.87 \pm 0.21$ | $\mathbf{3.93 \pm 0.22}$ |
| MelStyle-SVC | 0.9701/0.8691 | $4.11 \pm 0.20$ | $4.03 \pm 0.23$ | $3.84 \pm 0.27$ |
| CrossAtt-SVC | **0.9738/0.8693** | $\mathbf{4.14 \pm 0.33}$ | $\mathbf{4.04 \pm 0.13}$ | $3.88 \pm 0.27$ |

We also obtained informal feedback from in-house listeners in non-blind tests, and many reported that they prefer our proposed CrossAtt-SVC model's timbre compared to the original So-VITS conversions, because the originals have a muffled quality with singing that is not well-enunciated and sounds a bit like humming. This implies that our system has potential for good expressiveness, though the slight drop in MOS in Table 4 may be due to artifacts such as pitch fluctuations and background noise. Feedback indicates that our proposed model excels at reconstructing the high-frequency components of the singing voice, leading to improved clarity of articulation and enhanced brightness and fullness of the singer's timbre. This also contributes to a higher similarity score compared to baseline So-VITS. However, the overall quality of our model's conversion is not as robust as the original So-VITS, with occasional artifacts due to discontinuities introduced by our high-resolution frame-by-frame speaker embeddings.

In summary, using mel-style embeddings and the further step of cross-attention result in improvements in similarity and clarity. Similarity is improved because the frame-wise speaker embeddings are informed by the melody and capture the singer's varying vocals; clarity is improved because the frame-wise embeddings are informed by the content (lyrics and melody) and thus improve pronunciation and distinguishing spectro-temporal details of singing techniques.

## 6 Conclusion

Improvements in content representations have fueled rapid advances in singing voice conversion systems, but more work needs to be done in finding accurate timbre representations. This paper presents a cross-attention mechanism to generate frame-wise speaker embeddings informed by linguistic and melody information to better reflect how one singer's timbre can vary over time with different singing techniques and vocal ranges. Subjective and objective evaluations show encouraging results, with our model improving on clarity and voice similarity.

## Acknowledgment

This research was supported by the National Science and Technology Council of Taiwan under Grant No. 109-2221-E-007-094-MY3 and 111-2634-F-002-023.

## References

[1] Z. Li, B. Tang, X. Yin, Y. Wan, L. Xu, C. Shen, and Z. Ma, "PPG-based singing voice conversion with adversarial representation learning,"

*Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7073–7077, 2021.

[2] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, pp. 12449–12460, 2020.

[3] W.-N. Hsu, B. Bolte, Y.-H. H. Tsai, K. Lakhotia, R. Salakhutdinov, and A. Mohamed, "HuBERT: Self-supervised speech representation learning by masked prediction of hidden units," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 3451–3460, 2021.

[4] K. Qian, Y. Zhang, H. Gao, J. Ni, C.-I. Lai, D. Cox, M. Hasegawa-Johnson, and S. Chang, "ContentVec: An improved self-supervised speech representation by disentangling speakers," *Proceedings of the International Conference on Machine Learning (ICML)*, vol. 162, pp. 18003–18017, 2022.

[5] M. Morise, F. Yokomori, and K. Ozawa, "WORLD: A vocoder-based high-quality speech synthesis system for real-time applications," *IEICE Transactions on Information and Systems*, vol. E99-D, no. 7, pp. 1877–1884, 2016.

[6] J. W. Kim, J. Salamon, P. Li, and J. P. Bello, "CREPE: A convolutional representation for pitch estimation," *Proc. ICASSP*, pp. 161–165, 2018.

[7] H. Wei, X. Cao, T. Dan, and Y. Chen, "RMVPE: A robust model for vocal pitch estimation in polyphonic music," *Proceedings of Interspeech*, pp. 5421–5425, 2023.

[8] Y. Cho, Y. Tsao, H. Wang, and Y. Liu, "Mandarin singing voice synthesis with denoising diffusion probabilistic wasserstein gan," *Proceedings of the Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pp. 1956–1963, 2022.

[9] E. M. Liu, J.-W. Yeh, J.-H. Lu, and Y.-W. Liu, "Speaker embedding space cosine similarity comparisons of singing voice conversion models and voice morphing," *The Journal of the Acoustical Society of America*, vol. 154, p. A244, 2023.

[10] 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," *Proc. ICML*, vol. 162, pp. 2709–2720, 2022.

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

[12] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-Vectors: Robust DNN embeddings for speaker recognition," *Proc. ICASSP*, pp. 5329–5333, 2018.

[13] W.-C. Huang, L. P. Violeta, S. Liu, J. Shi, and T. Toda, "The singing voice conversion challenge 2023," *Proceedings of the IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pp. 1–8, 2023.

[14] J. Kong, J. Kim, and J. Bae, "HiFi-GAN: Generative adversarial networks for efficient and high fidelity speech synthesis," *Proc. NeurIPS*, vol. 33, pp. 17022–17033, 2020.

[15] D. Min, D. B. Lee, E. Yang, and S. J. Hwang, "Meta-StyleSpeech : Multi-speaker adaptive text-to-speech generation," *Proc. ICML*, vol. 139, pp. 7748–7759, 2021.

[16] L. Zhang, R. Li, S. Wang, L. Deng, J. Liu, Y. Ren, J. He, R. Huang, J. Zhu, X. Chen, and Z. Zhao, "M4Singer: A multi-style, multi-singer and musical score provided mandarin singing corpus," *Proc. NeurIPS, Datasets and Benchmarks Track*, 2022.

[17] L. McInnes, J. Healy, N. Saul, and L. Großberger, "UMAP: Uniform manifold approximation and projection," *Journal of Open Source Software*, vol. 3, no. 29, p. 861, 2018.

[18] A. Nagrani, J. S. Chung, W. Xie, and A. Zisserman, "Voxceleb: Large-scale speaker verification in the wild," *Computer Speech Language*, vol. 60, p. 101027, 2020.

[19] D. van der Vloed, F. Kelly, and A. Alexander, "Exploring the effects of device variability on forensic speaker comparison using vocalise and nfi-

frida, a forensically realistic database," *Proceedings of The Speaker and Language Recognition Workshop*, pp. 402–407, 2020.

[20] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Proc. NeurIPS*, vol. 30, pp. 5998–6008, 2017.

[21] C.-C. Chu, F.-R. Yang, Y.-J. Lee, Y.-W. Liu, and S.-H. Wu, "MPop600: A mandarin popular song database with aligned audio, lyrics, and musical scores for singing voice synthesis," *Proc. APSIPA ASC*, pp. 1647–1652, 2020.

[22] R. Huang, F. Chen, Y. Ren, J. Liu, C. Cui, and Z. Zhao, "Multi-Singer: Fast Multi-Singer singing voice vocoder with a large-scale corpus," *Proceedings of the ACM International Conference on Multimedia*, pp. 3945–3954, 2021.

[23] M. T. Akhter, P. Banerjee, S. Dhar, and N. D. Jana, "An analysis of performance evaluation metrics for voice conversion models," *Proceedings of the IEEE India Council International Conference*, pp. 1–6, 2022.

[24] L. Wan, Q. Wang, A. Papir, and I. L. Moreno, "Generalized end-to-end loss for speaker verification," *Proc. ICASSP*, pp. 4879–4883, 2018.