Many-to-many Singing Performance Style Transfer on Pitch and Energy Contours

Yu-Teng Hsu, Jun-You Wang, and Jyh-Shing Roger Jang

***Abstract*—Singing voice conversion (SVC) aims to convert the singer identity of a singing voice to that of another singer. However, most existing SVC systems only perform the conversion of timbre information, while leaving other information unchanged. This approach does not consider other aspects of singer identity, particularly a singer's performance style, which is reflected in the pitch (F0) and the energy (volume dynamics) contours of singing. To address this issue, this paper proposes a many-to-many singing performance style transfer system that converts the pitch and energy contours of one singer's style to another singer's. To achieve this target, we utilize two AutoVC-like autoencoders with an information bottleneck to automatically disentangle performance style from other musical contents, one for the pitch contour while another for the energy contour. Experiment results suggested that the proposed model can perform singing performance style transfer in a many-to-many conversion scenario, resulting in improved singer identity similarity to the target singer.**

***Index Terms*—Singing style transfer, singing voice conversion.**

I. Introduction

Singing voice conversion (SVC) aims to alter the *singer identity* of a singing voice to that of another singer while keeping the *musical content* unchanged. Previous SVC systems [1]–[7] have mainly focused on modifying the *timbre* of the voice to achieve this goal. However, this approach simplifies *singer identity* to the pure *timbre* information and does not consider the *performance style*. Consequently, while listeners may agree that the audio converted by these SVC systems sounds like the target singer's voice, it does not reflect *how* the target singer would perform the musical piece.

The underlying problem is that different singers have their own interpretations of the same musical piece, which affects the musical expression of the performance [8]. For example, a singer may choose to use *vibrato* when singing a long note or may choose not to. These choices shape the unique performance style of a singer, which is identifiable by both humans and machine learning algorithms [9]. In this paper, we are particularly interested in the performance style manifested in the pitch (fundamental frequency, f_0) and energy (volume dynamics) contours of singing, as they are important for identifying styles in both singing [9] and instrumental performances [10]. The task of predicting pitch and energy contours to synthesize expressive music performances has also been studied (pitch only: [11]–[13]; energy only: [10]; both: [14]–[16]). Therefore, during SVC, it is desirable to consider the transfer of performance styles in terms of pitch and energy contours. Note that none of the aforementioned works focus on style transfer (converting existing pitch/energy contours to another performer's style), but rather on music synthesis (predicting expressive pitch/energy contours from a musical score). These synthesis tasks require a transcribed musical score, whereas performance style transfer does not, which enables applications when the musical score is not available.

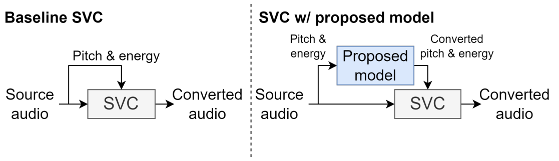

Fig. 1. The application scenario of the proposed model. Our model can be combined with a previous SVC model to achieve the conversion of both timbre and performance style (in terms of pitch and energy contours).

To address this issue, we propose a system that performs *singing performance style transfer* on pitch and energy contours in a many-to-many scenario. Our system accepts pitch and energy contours as inputs and converts their performance styles to that of a specific target singer. This problem definition encompasses the use of vibratos, overshoots, preparations [17], glissandi, and other singing techniques identifiable in pitch contours, as well as the volume dynamics observable in energy contours. To the best of our knowledge, this is the first work that addresses singing performance style transfer on pitch and energy contours. This system can be integrated with any of the previous SVC systems to achieve a better conversion in terms of overall singer identity, as illustrated in Figure 1.

Drawing inspiration from AutoVC [18], we employ autoencoders with an information bottleneck to disentangle performance style from musical content within pitch and energy contours (e.g., note pitches and the dynamics caused by pronouncing different phonemes). The proposed system contains two autoencoders: one for converting the pitch contour and the other for the energy contour. Experiment results show that integrating an SVC model with the proposed style transfer model further improves the perceptual similarity of the converted audio to the target singer's singer identity. The source code is available at https://github.com/poohhsu/Singing-Performance-Style-Transfer.

II. Methods

The proposed singing performance style transfer system consists of two main components: a pitch conversion model and an energy conversion model, which are optimized separately. An overview of the proposed system is shown in Figure 2. The system takes as input a pitch contour $\mathbf{p} \in \mathbb{R}^m$ and a log-scale energy contour $\mathbf{e} \in \mathbb{R}^m$, both with m frames and a sampling rate of 200 Hz, along with a target singer ID.

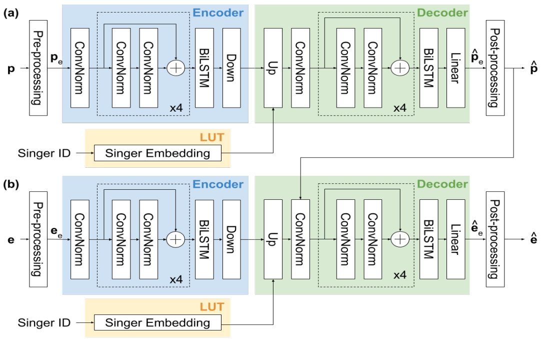

Fig. 2. The overview of the proposed system. *Down* and *Up* denote downsampling and upsampling, respectively. During inference, we first run the pitch conversion model (a), and then the energy conversion model (b) in a cascading manner.

It then converts the performance styles of $\mathbf{p}$ and $\mathbf{e}$ to those of the target singer. This system operates in a many-to-many scenario, meaning it can convert $\mathbf{p}$ and $\mathbf{e}$ to the performance styles of a fixed set of singers seen during model training.

A. Pitch Conversion

As shown in Figure 2 (a), the proposed pitch conversion model is an AutoVC-like autoencoder with an information bottleneck [18]. It consists of an encoder, a decoder, and a trainable singer embedding lookup table (LUT). The model takes $\mathbf{p}$ and a target singer ID as input for style transfer.

Pre-processing. We first convert $\mathbf{p}$ to a pitch embedding $\mathbf{p}_{\text{e}} \in \mathbb{R}^{m\times72}$, where each pitch value is represented by a 72-dimensional vector. Each element of the vector corresponds to an integer MIDI number from C1 (32.7 Hz) to B6 (1975.5 Hz). For pitch values that do not equal integer MIDI numbers, we use linear interpolation to calculate their embeddings. This pre-processing is applied to each frame of $\mathbf{p}$ to obtain $\mathbf{p}_{\text{e}}$.

Encoder. The encoder processes the pitch embedding $\mathbf{p}_{\text{e}}$ with a 1-D convolution layer with a kernel size of 11. The output is then passed through 4 residual blocks [19], each comprising 2 1-D convolution layers with a kernel size of 5. Each convolutional layer is followed by group normalization [20] and a ReLU activation function, denoted as *ConvNorm* in Figure 2 (a). The output is then fed into a BiLSTM layer with an output dimension of 2 for each direction. Finally, a downsampling operation is performed to form an information bottleneck, with a downsampling rate of 128. This results in an information bottleneck of 6.25 dimensions per second.

Decoder. The decoder takes both the encoding produced by the encoder and a singer embedding from the LUT as input. It upsamples both inputs to match the original frame number of $\mathbf{p}_{\text{e}}$ and concatenates them. Similar to the encoder, the decoder processes the concatenated features with a 1-D convolution layer, 4 residual blocks, and a BiLSTM layer. Finally, a linear layer is applied to generate the output $\hat{\mathbf{p}}_{\text{e}} \in \mathbb{R}^{m\times72}$.

Post-processing. To convert $\hat{\mathbf{p}}_{\text{e}}$ back to the pitch contour prediction $\hat{\mathbf{p}} \in \mathbb{R}^{m}$, following CREPE [21], we calculate a weighted average of the associated MIDI number of each element according to the output $\hat{\mathbf{p}}_{\text{e}}$.

Loss functions. Similar to AutoVC [18], the model is trained to reconstruct the original input ($\mathbf{p}$ and $\mathbf{p}_{\text{e}}$) by feeding the corresponding singer ID as the target singer ID. The loss function consists of two main components. The first is the pitch reconstruction loss $\mathcal{L}^{\text{p}}_{\text{Recon}}$, defined as:

$$\begin{aligned}\mathcal{L}^{\text{p}}_{\text{ReconE}} &= \text{BCE}(\mathbf{p}_{\text{e}}, \hat{\mathbf{p}}_{\text{e}}),\\ \mathcal{L}^{\text{p}}_{\text{ReconC}} &= \text{RMSE}(\mathbf{p}, \hat{\mathbf{p}}),\\ \mathcal{L}^{\text{p}}_{\text{Recon}} &= \lambda^{\text{p}}_{\text{ReconE}}\mathcal{L}^{\text{p}}_{\text{ReconE}} + \lambda^{\text{p}}_{\text{ReconC}}\mathcal{L}^{\text{p}}_{\text{ReconC}},\end{aligned} \tag{1}$$

where BCE denotes the binary cross-entropy loss, and RMSE denotes the root mean square error loss. The superscript p indicates that these loss terms are used to train the pitch conversion model. The weighting factors $\lambda^{\text{p}}_{\text{ReconE}}$ and $\lambda^{\text{p}}_{\text{ReconC}}$ are empirically set to 1 and 10, respectively.

Furthermore, to encourage the model to learn vibrato-related features, which are particularly important for performance styles [9], [11], [13]. we include the second component, namely the vibrato loss $\mathcal{L}^{\text{p}}_{\text{Vib}}$, in our loss function. Following [22], given the ground-truth pitch contour $\mathbf{p}$ and the predicted pitch contour $\hat{\mathbf{p}}$, we first perform a sharpening operation on them to isolate vibratos (similar to [11], [22]). We then extract two sets of features from the sharpened pitch contours for computing loss functions. The first is their corresponding Short-time Fourier Transform (STFT) power spectrograms (denoted as $\mathbf{p}_{\text{FT}}$ and $\hat{\mathbf{p}}_{\text{FT}}$, respectively). The second is the vibrato extent (amplitude) contours $\mathbf{p}_{\text{VibExt}}$ and $\hat{\mathbf{p}}_{\text{VibExt}}$ that represent frame-wise vibrato amplitudes of the pitch contours. The methods to extract these features are similar to [22], detailed as follows:

$$\begin{aligned}\mathbf{p}_{\text{sharp}} &= \mathbf{p} - \text{sinc}(i) * \mathbf{p},\\ \mathbf{p}_{\text{FT}} &= \text{STFT}(\mathbf{p}_{\text{sharp}}),\\ \mathbf{p}_{\text{VibExt}} &= \text{vib\_frame\_max}(\mathbf{p}_{\text{FT}}),\end{aligned} \tag{2}$$

where $*$ denotes convolution, i denotes the frame index, sinc denotes the sinc function, vib_frame_max denotes the frame-wise maximum operation applied to the frequency bins between 5 and 8 Hz, which is the typical frequency range of vibratos. Similarly, we obtain $\hat{\mathbf{p}}_{\text{VibExt}}$ and $\hat{\mathbf{p}}_{\text{FT}}$ from $\hat{\mathbf{p}}$. The vibrato loss is then computed as follows:

$$\begin{aligned}\mathcal{L}^{\text{p}}_{\text{FT}} &= \text{RMSE}(\mathbf{p}_{\text{FT}}, \hat{\mathbf{p}}_{\text{FT}}),\\ \mathcal{L}^{\text{p}}_{\text{VibExt}} &= \text{RMSE}(\mathbf{p}_{\text{VibExt}}, \hat{\mathbf{p}}_{\text{VibExt}}),\\ \hat{\mathbf{p}}_{\text{IsVib}} &= \text{Thresholding}(\hat{\mathbf{p}}_{\text{VibExt}}, \text{thres}_{\text{p}}),\\ \hat{\mathbf{p}}_{\text{VibDiff}} &= \text{Diff}(\hat{\mathbf{p}}_{\text{VibExt}}) \times \hat{\mathbf{p}}_{\text{IsVib}},\\ \mathcal{L}^{\text{p}}_{\text{Smooth}} &= \text{RMS}(\hat{\mathbf{p}}_{\text{VibDiff}}),\\ \mathcal{L}^{\text{p}}_{\text{Vib}} &= \lambda^{\text{p}}_{\text{FT}}\mathcal{L}^{\text{p}}_{\text{FT}} + \lambda^{\text{p}}_{\text{VibExt}}\mathcal{L}^{\text{p}}_{\text{VibExt}} + \lambda^{\text{p}}_{\text{Smooth}}\mathcal{L}^{\text{p}}_{\text{Smooth}},\end{aligned} \tag{3}$$

where Thresholding is a function that operates on each element of $\hat{\mathbf{p}}_{\text{VibExt}}$ except the last one. For index i, it outputs 1 if both the i-th element $\hat{\mathbf{p}}^{(i)}_{\text{VibExt}}$ and the $(i{+}1)$-th element $\hat{\mathbf{p}}^{(i+1)}_{\text{VibExt}}$ are larger than a given threshold thres_{p}; otherwise, it outputs 0. We set thres_{p} to 0.75, which determines whether a frame contains vibrato. Diff denotes first-order difference, and RMS denotes the root mean square function. The weighting factors $\lambda^{\text{p}}_{\text{FT}}$, $\lambda^{\text{p}}_{\text{VibExt}}$, and $\lambda^{\text{p}}_{\text{Smooth}}$ are all set to 0.1 empirically.

Finally, the total loss $\mathcal{L}^{\text{p}}$ is set to the sum of the pitch reconstruction loss $\mathcal{L}^{\text{p}}_{\text{Recon}}$ and the vibrato loss $\mathcal{L}^{\text{p}}_{\text{Vib}}$.

B. Energy Conversion

The proposed energy conversion model is depicted in Figure 2(b). It takes as input the energy contour $\mathbf{e}$, a target singer ID, and the pitch contour $\hat{\mathbf{p}}$, which is obtained by converting $\mathbf{p}$ to the target singer's style using the pitch conversion model described earlier. The model then converts the performance style of $\mathbf{e}$ to that of the target singer. In other words, the proposed model works in a cascading manner during inference time. During training, we provide the ground-truth pitch contour $\mathbf{p}$ to the energy conversion model as side information.

Pre-processing. We convert the input log-scale energy contour $\mathbf{e}$ into an energy embedding $\mathbf{e}_\mathrm{e} \in \mathbb{R}^{m\times 128}$. Specifically, we define the energy range of interest from 10^{-4} to 1 and divide this range into 128 logarithmically spaced bins. We then apply linear interpolation to convert each value in $\mathbf{e}$ to this 128-dimensional embedding.

Model architecture. The encoder architecture is identical to that of the pitch conversion model. As for the decoder, the only difference is that we concatenate $\hat{\mathbf{p}}$ with other features after the upsampling layer, as illustrated in Figure 2(b).

Loss functions. The loss functions are the same as those used for the pitch conversion model (see Equation 1, 2, and 3 for more details). The weighting factors for the energy loss $\mathcal{L}^\mathrm{e}$, $\lambda^\mathrm{e}_\mathrm{ReconE}$, $\lambda^\mathrm{e}_\mathrm{ReconC}$, $\lambda^\mathrm{e}_\mathrm{FT}$, $\lambda^\mathrm{e}_\mathrm{VibExt}$, and $\lambda^\mathrm{e}_\mathrm{Smooth}$, are empirically set to 1, 10, 0.01, 0.01, and 0.01, respectively. The threshold $\mathrm{thres}_\mathrm{e}$ that determines the presence of vibratos is set to 20.

III. Experiment Setup

A. Datasets

We use three datasets in our experiments: M4Singer [23], Opencpop [24], and TONAS [25]. M4Singer contains 29.8 hours of Mandarin pop music sung by 20 singers. Opencpop consists of 5.2 hours of Mandarin pop music sung by one singer. TONAS includes 0.34 hours of *a cappella* singing in the Flamenco style. In total, our training data comprises 22 singers (20 from M4Singer, 1 from Opencpop, and 1 from TONAS). For data partitioning, we use an 8:1:1 partition at the song level for M4Singer and TONAS. For Opencpop, we follow the official train/test split and divide the training data into training and validation sets with an 8:1 ratio at song level.

B. Implementation Details

We resample audios to 16 kHz and use CREPE [21] to extract pitch contours with a window size of 1024 and a hop size of 80. The same parameters are used to compute energy contours. We set the dimensionality of all hidden layers and singer embeddings to 128. The model is trained for 400,000 steps with a batch size of 16, using the AdamW optimizer with a learning rate of 10^{-4} and a weight decay of 10^{-4}. During training, we apply data augmentation by randomly transposing the pitch contour within the range of C1 and B6.

C. Evaluation Methods and Baselines

We conducted both objective and subjective evaluations in the experiments. For the objective test, we trained two modified ResNetSE34L speaker verification systems [26] for singer verification on pitch and energy contours, respectively. We changed the input from waveforms to pitch/energy embeddings and replaced all 2D convolutions with 1D convolutions. We used the training sets of M4Singer, Opencpop, and TONAS to train the two models from scratch with the AM-Softmax loss [27]. During evaluation, we used the trained models to extract fixed-dimensional singer embeddings from pitch/energy contours. We then computed the cosine similarity between the embeddings of the converted pitch/energy contours and the average singer embedding of the target singer's pitch/energy contours extracted by the same model. A higher similarity means that the pitch/energy contours are more similar to the target singer's style.

For the subjective test, we focus on the effectiveness of the proposed style transfer model when integrated into an SVC model, which we consider the main application scenario of the proposed model. We employ Diff-SVC (https://github.com/prophesier/diff-svc), a popular any-to-one SVC model based on diffusion mechanisms, in the evaluation. In each test, we compare two settings: 1) using Diff-SVC for timbre conversion without any modification (denoted as **Baseline SVC**), and 2) using Diff-SVC for timbre conversion while replacing the pitch and energy contours with those converted by (one of) our performance style transfer model to the target singer's style. Figure 1 shows the comparison of the two settings.

Since Diff-SVC is an any-to-one SVC model, we selected four target singers (Opencpop's only singer, M4Singer's Alto-1, Tenor-3, and Tenor-7) to train their own Diff-SVC model. We then conducted a comparative mean opinion score (CMOS) [28] test. For each question, participants were asked to listen to two converted audio clips and one clip of the target singer. Then, they were asked to rate their relative preference between the two converted audios based on **naturalness** and **similarity** to the performance style of the target singer. The CMOS scale ranges from -3 to 3.

Baseline models: As there is no previous work with the same problem definition, we created three more baselines for comparison, including 1) **Source**: the unconverted (source) pitch/energy contours, which serve as a lower bound; 2) **Proposed w/o cascade**: an ablation of the proposed model where the converted pitch contour is not provided to the energy conversion model (i.e., not cascading the two models), which may cause inconsistency between the two converted contours; and 3) **Vib-scaling**: a simple baseline that computes the vibrato statistics (mean and variance) of each singer in the training data and uses these statistics to scale the vibrato amplitudes of the pitch/energy contours during inference.

IV. Results

A. Objective Results

In the objective test, we evaluated three models: **Proposed**, **Proposed w/o cascade**, and **Vib-scaling**. Additionally, we used the target singer's pitch and energy contours as an upper bound, denoted as **Target**. Specifically, we randomly selected 2 audio clips for each source-target pair, which leads to $22 \times 21 \times 2 = 924$ audio clips.

Table I presents the objective results. Compared to **Source**, **Proposed** improved singer embedding similarity in both pitch

TABLE I
OBJECTIVE EXPERIMENT RESULTS IN TERMS OF COSINE SIMILARITY.

Model	Pitch	Energy
Source	0.226	0.295
Vib-scaling	0.283	0.292
Proposed w/o cascade	0.435	0.402
Proposed	0.435	0.414
Target	**0.555**	**0.681**

TABLE II
A DETAILED VIEW OF OBJECTIVE EXPERIMENT RESULTS. THE SINGERS ARE SEPARATED TO "FLA" (FLAMENCO) AND "MAN" (MANDARIN POP) BASED ON THEIR SPECIFIC PERFORMANCE STYLES, WITH THE OBJECTIVE SIMILARITIES BEING REPORTED SEPARATELY.

Model	Pitch		Energy	
	Fla↔Man	Man↔Man	Fla↔Man	Man↔Man
Source	-0.054	0.254	0.191	0.306
Proposed	**0.478**	**0.431**	**0.333**	**0.423**

and energy contours, indicating that the proposed model is capable of performing performance style transfer. However, a clear difference remains between **Proposed** and **Target**, suggesting that there is still much room for improvement. In comparing **Proposed** with **Proposed w/o cascade**, **Proposed** slightly outperformed **Proposed w/o cascade** in energy contour similarity, showing the effectiveness of incorporating pitch contours into the energy conversion model. Finally, **Proposed** outperformed **Vib-scaling** considerably. This indicates that even with the added loss term to emphasize vibrato components, the proposed system did not reduce to only modifying vibratos (though it does modify vibratos; see Section IV-C). Instead, it achieved better performance by considering and converting various aspects of performance styles.

To examine the performance of the proposed model under different degrees of performance style transfer, we further computed separate objective results on performance style transfer for two cases: 1) between the Flamenco singer (TONAS) and Mandarin pop singers (M4Singer and Opencpop), and 2) within Mandarin pop singers. The former represents a larger degree of style transfer, while the latter represents a smaller degree of style transfer. Table II shows that in both cases, **Proposed** is capable of achieving improved objective similarities.

B. *Subjective Results*

We conducted two subjective tests: one comparing **Baseline SVC** with **Proposed**, and the other comparing **Baseline SVC** with **Proposed w/o cascade**. We recruited 10 participants, each of whom was asked to rate 12 audio pairs anonymously. All participants understood and agreed that their responses would be used solely for this research. The results are presented in Table III. Regarding singer identity similarity, **Proposed** outperformed **Baseline SVC** significantly ($p \approx 0.017$), indicating that the proposed model leads to improved perceptual singer similarity. Interestingly, although **Proposed w/o cascade** outperformed **Source** in the objective test, it performed significantly worse than **Baseline SVC** in subjective similarity ($p \approx 0.040$). We suspect that the inconsistency between the pitch and energy contours in **Proposed w/o cascade** heavily affects human perception of performance style. As a result, human raters tend to perceive the singer identity of **Proposed w/o cascade** as dissimilar to that of the target singer.

TABLE III
SUBJECTIVE CMOS EXPERIMENT RESULTS.

Model	Similarity	Naturalness
Baseline SVC	0.00	**0.00**
Proposed w/o cascade	-0.25 ± 0.28	-1.31 ± 0.28
Proposed	**+0.34 ± 0.32**	-1.23 ± 0.29

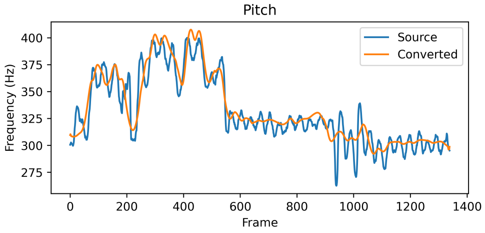

Fig. 3. A pitch conversion example using **Proposed**. The amplitude of vibratos is largely reduced, which aligns with the performance style of the target singer `Opencpop`, who seldom sings vibratos with large amplitudes.

Regarding naturalness, **Baseline SVC** significantly outperformed both **Proposed** and **Proposed w/o cascade** (both $p < 10^{-13}$). This reflects a side effect of modifying the pitch and energy contours, which leads to a degradation in naturalness. Future work could focus on alleviating this issue.

C. *Case Study*

To understand how the proposed performance style transfer model actually does, we visualize an example in Figure 3. In this case, the source audio from the TONAS dataset contains clear vibratos, a characteristic feature of Flamenco singing. The target singer from the Opencpop dataset, however, seldom uses vibratos with large amplitudes. The proposed model successfully captured this difference in performance style and modified the pitch contour to reduce the amplitude of vibratos. As a result, the performance style of the converted pitch contour aligns better with that of `Opencpop`. This is also reflected in the objective result, as the pitch contour similarity to `Opencpop`'s average singer embedding increases from -0.205 (source) to 0.146 (converted using **Proposed**).

V. CONCLUSION

We propose the first many-to-many singing performance style transfer system that focuses on the style transfer of pitch and energy contours. The task is decomposed into pitch conversion and energy conversion, with two models trained to address these aspects. Using an AutoVC-like architecture and incorporating pitch contour as side information for the energy conversion model, experimental results demonstrate that the proposed model can successfully perform style transfer in a many-to-many scenario. For future work, we plan to adopt other disentangled representation learning methods and increase the dataset scale to further improve performance.

References

[1] C. Deng, C. Yu, H. Lu, C. Weng, and D. Yu, "Pitchnet: Unsupervised singing voice conversion with pitch adversarial network," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 7749–7753.

[2] S. Nercessian, "Zero-shot singing voice conversion." in *ISMIR*, 2020, pp. 70–76.

[3] S. Liu, Y. Cao, N. Hu, D. Su, and H. Meng, "Fastsvc: Fast cross-domain singing voice conversion with feature-wise linear modulation," in *2021 ieee international conference on multimedia and expo (icme)*, 2021, pp. 1–6.

[4] S. Liu, Y. Cao, D. Su, and H. Meng, "Diffsvc: A diffusion probabilistic model for singing voice conversion," in *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2021, pp. 741–748.

[5] Y. Lu, Z. Ye, W. Xue, X. Tan, Q. Liu, and Y. Guo, "Comosvc: Consistency model-based singing voice conversion," *arXiv preprint arXiv:2401.01792*, 2024.

[6] B. Sha, X. Li, Z. Wu, Y. Shan, and H. Meng, "Neural concatenative singing voice conversion: Rethinking concatenation-based approach for one-shot singing voice conversion," *CoRR*, vol. abs/2312.04919, 2023.

[7] W. Huang, L. P. Violeta, S. Liu, J. Shi, and T. Toda, "The singing voice conversion challenge 2023," in *IEEE Automatic Speech Recognition and Understanding Workshop, ASRU 2023, Taipei, Taiwan*, 2023, pp. 1–8.

[8] G. G. Xia and S. Dai, "Music style transfer: A position paper," in *Proceedings of the 6th International Workshop on Musical Metacreation, Salamanca, Spain*, 2018. [Online]. Available: https://musicalmetacreation.org/mume2018/proceedings/Xia.pdf

[9] T. Kako, Y. Ohishi, H. Kameoka, K. Kashino, and K. Takeda, "Automatic identification for singing style based on sung melodic contour characterized in phase plane," in *Proceedings of the 10th International Society for Music Information Retrieval Conference, ISMIR 2009, Kobe, Japan*, 2009, pp. 393–398.

[10] G. Widmer and W. Goebl, "Computational models of expressive music performance: The state of the art," *Journal of New Music Research*, vol. 33, no. 3, pp. 203–216, 2004.

[11] L. Ardaillon, C. Chabot-Canet, and A. Roebel, "Expressive control of singing voice synthesis using musical contexts and a parametric F0 model," in *17th Annual Conference of the International Speech Communication Association, Interspeech 2016, San Francisco, CA, USA*, 2016, pp. 1250–1254.

[12] Y. Ohishi, H. Kameoka, D. Mochihashi, and K. Kashino, "A stochastic model of singing voice F0 contours for characterizing expressive dynamic components," in *13th Annual Conference of the International Speech Communication Association, INTERSPEECH 2012, Portland, Oregon, USA*, 2012, pp. 474–477.

[13] Y. Ikemiya, K. Itoyama, and H. G. Okuno, "Transferring vocal expression of F0 contour using singing voice synthesizer," in *Modern Advances in Applied Intelligence - 27th International Conference on Industrial Engineering and Other Applications of Applied Intelligent Systems, IEA/AIE 2014, Kaohsiung, Taiwan*, 2014, pp. 250–259.

[14] N. Jonason, B. Sturm, and C. Thomé, "The control-synthesis approach for making expressive and controllable neural music synthesizers," in *2020 AI Music Creativity Conference*, 2020. [Online]. Available: https://boblsturm.github.io/aimusic2020/papers/CSMC__MuMe_2020_paper_29.pdf

[15] T. Nakano and M. Goto, "Vocalistener: A singing-to-singing synthesis system based on iterative parameter estimation," in *Proceedings of SMC 2009*, 2009, pp. 343–348.

[16] Y. Song, W. Song, W. Zhang, Z. Zhang, D. Zeng, Z. Liu, and Y. Yu, "Singing voice synthesis with vibrato modeling and latent energy representation," in *2022 IEEE 24th International Workshop on Multimedia Signal Processing (MMSP)*, 2022, pp. 1–6.

[17] T. Saitou, M. Unoki, and M. Akagi, "Extraction of f0 dynamic characteristics and development of f0 control model in singing voice," in *Proc. ICAD*, 2002, pp. 275–278.

[18] 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*, 2019, pp. 5210–5219.

[19] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[20] Y. Wu and K. He, "Group normalization," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 3–19.

[21] J. W. Kim, J. Salamon, P. Li, and J. P. Bello, "Crepe: A convolutional representation for pitch estimation," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 161–165.

[22] R. Liu, X. Wen, C. Lu, L. Song, and J. S. Sung, "Vibrato learning in multi-singer singing voice synthesis," in *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2021, pp. 773–779.

[23] L. Zhang, R. Li, S. Wang, L. Deng, J. Liu, Y. Ren, J. He, R. Huang, J. Zhu, X. Chen *et al.*, "M4singer: A multi-style, multi-singer and musical score provided mandarin singing corpus," *Advances in Neural Information Processing Systems*, vol. 35, pp. 6914–6926, 2022.

[24] Y. Wang, X. Wang, P. Zhu, J. Wu, H. Li, H. Xue, Y. Zhang, L. Xie, and M. Bi, "Opencpop: A high-quality open source chinese popular song corpus for singing voice synthesis," *arXiv preprint arXiv:2201.07429*, 2022.

[25] J. Mora, F. Gómez, E. Gómez, F. Escobar, and J. M. Díaz-Báñez, "Melodic characterization and similarity in a cappella flamenco cantes," in *International Society for Music Information Retrieval Conference (ISMIR)*, 2010. [Online]. Available: https://mtg.upf.edu/node/1708

[26] J. S. Chung, J. Huh, S. Mun, M. Lee, H. S. Heo, S. Choe, C. Ham, S. Jung, B.-J. Lee, and I. Han, "In defence of metric learning for speaker recognition," *arXiv preprint arXiv:2003.11982*, 2020.

[27] F. Wang, J. Cheng, W. Liu, and H. Liu, "Additive margin softmax for face verification," *IEEE Signal Processing Letters*, vol. 25, no. 7, pp. 926–930, 2018.

[28] P. C. Loizou, "Speech quality assessment," in *Multimedia analysis, processing and communications*. Springer, 2011, pp. 623–654.