

LDM-SVC: Latent Diffusion Model Based Zero-Shot Any-to-Any Singing Voice Conversion with Singer Guidance

Shihao Chen^{1,2}, Yu Gu², Jie Zhang¹, Na Li², Rilin Chen², Liping Chen¹, Lirong Dai¹

¹NERC-SLIP, University of Science and Technology of China (USTC), China

² Tencent AI Lab

shchen16@mail.ustc.edu.cn, colinygu@tencent.com, jzhang6@ustc.edu.cn

Abstract

Any-to-any singing voice conversion (SVC) is an interesting audio editing technique, aiming to convert the singing voice of one singer into that of another, given only a few seconds of singing data. However, during the conversion process, the issue of timbre leakage is inevitable: the converted singing voice still sounds like the original singer’s voice. To tackle this, we propose a latent diffusion model for SVC (LDM-SVC) in this work, which attempts to perform SVC in the latent space using an LDM. We pretrain a variational autoencoder structure using the noted open-source So-VITS-SVC project based on the VITS framework, which is then used for the LDM training. Besides, we propose a singer guidance training method based on classifier-free guidance to further suppress the timbre of the original singer. Experimental results show the superiority of the proposed method over previous works in both subjective and objective evaluations of timbre similarity.

Index Terms: Singing voice conversion, latent diffusion model, variational autoencoder, classifier-free guidance

1. Introduction

Singing Voice Conversion (SVC) is a popular audio editing technique that aims to change the singing voice of one singer to mimic another. Different from singing voice synthesis which requires well-designed musical note inputs [1, 2], this technique allows users to customize their favorite singers performing any songs just given corresponding recorded songs sung by other singers. Unlike the many-to-many or many-to-one scenarios, the any-to-any SVC is much more challenging, which demands the model to perform conversion for any target singer who were not included in the training set by solely a short snippet of reference singing voice that even lasts for few seconds.

The main challenge of SVC is to separate and reassemble the singer’s unique vocal timbre from the content and melody of songs. Similarly to voice conversion, mainstreaming SVC systems also follows a recognition-synthesis scheme as a typical two-stage process. In the first stage, singer-independent features such as phonetic posteriorgrams (PPG) [3–7] from an ASR model and self-supervised learning (SSL) representations [8, 9] trained on large amounts of unlabeled speech data are used to encode audio. These representations serve as intermediary for SVC, which can effectively extract content and semantic information from waveforms. In the second stage, acoustic models are involved to generate the target audio or acoustic fea-

tures from these immediate representations. Various generative models have been employed for SVC decoding, including autoregressive models [10–13], generative adversarial networks (GANs) [4, 5, 14], variational autoencoder (VAE) [15] and diffusion models [6]. Despite of naturalness, sound quality and intonation accuracy of converted singing voice have largely improved by above different SVC models, the timbre leakage problems remain serious, especially for the challenges of SVC cross different genders. This is primarily due to PPG and SSL features containing not only content information but also some timbre information of the original singer. To alleviate the timbre leakage, many works such as So-VITS-SVC¹ involved information perturbation by directly adding white noises on hidden features or acoustic features. However such white noises were totally independent with singer information and the information perturbation modules were not trainable and optimized in the network training stage and directly adding noise on acoustic features may also lead in pronunciation and quality distortion.

Recently, Latent Diffusion Model (LDM) based systems have shown a great success in image generation from text such as Stable Diffusion [16] and high-quality sound generation from text such as Tango [17] and AudioLDM [18, 19], which performed forward and denoising diffusion processes on the hidden spaces rather than acoustic features like other models [20, 21]. Motivated by these models, we present LDM-SVC, a novel any-to-any SVC method which reconstructs the waveform directly from the latent representation in an end-to-end latent diffusion manner. Unlike DiffSVC [6] employed on mel-spectrograms and requiring an additional vocoder, we conduct the diffusion model on the hidden space from a pre-trained VAE model using So-VITS-SVC and directly utilize the predicted latent representation to generate the waveforms by the VAE decoder. To address the timbre leakage issue, we regard the LDM forward process as a information perturbation process in those both progressively adding noise to decouple the singer timbre from content and melody. Different from the aforementioned methods which simply add white noise on waveforms, such information perturbation module is trainable and conditioned on singer information. To better decouple singer information, a singer guidance training mechanism is explored, which is inspired by the classifier-free method [22] in image generation when training the conditional and unconditional diffusion model at the same time. Comparing with many state-of-the-art SVC models, both subjective and objective experimental results indicate that our proposed method can achieve greater timbre similarity in any-to-any SVC tasks for both seen singer conversion and unseen singer conversion scenarios and better singing naturalness.

This work was done at Tencent AI Lab as an internship by Shihao Chen. This work was supported by the National Natural Science Foundation of China (62101523), Hefei Municipal Natural Science Foundation (2022012) and USTC Research Funds of the Double First-Class Initiative (YD2100002008). (Correspondence: jzhang6@ustc.edu.cn)

¹<https://github.com/PlayVoice/so-vits-svc-5.0/tree/bigvgan-mix-v2>

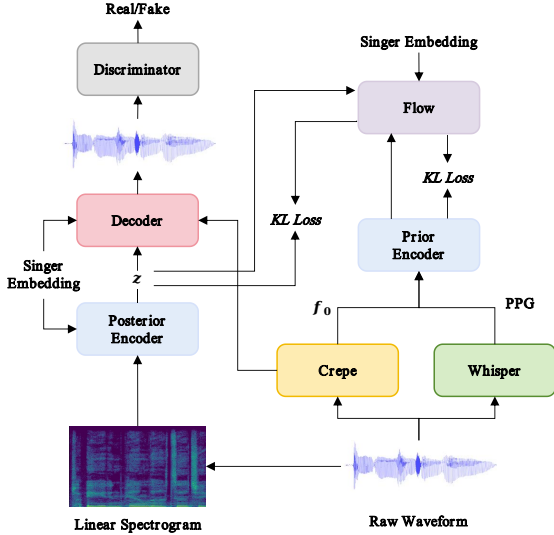


Figure 1: *Left: Pre-training procedure of So-VITS-SVC; Right: Training procedure of LDM-SVC.*

The rest of this paper is organized as follows. Section 2 outlines the proposed LDM-SVC method. Experiments are presented in Section 3. Finally, Section 4 concludes this work.

2. Proposed Method

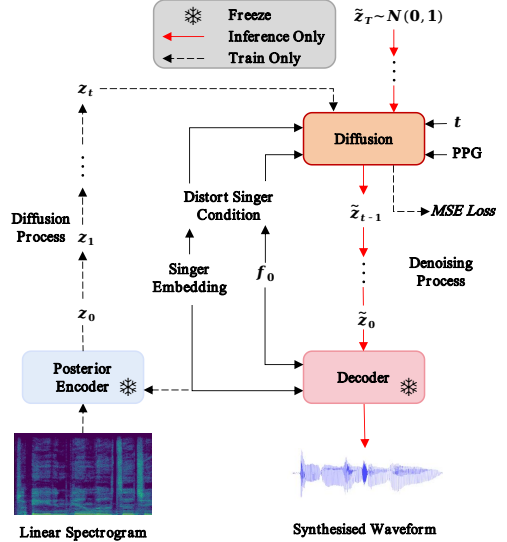
2.1. VAE Pretraining

We pretrain the VAE model using So-VITS-SVC, which is a VAE based any-to-any SVC model following the VITS framework [23] and consists of three key components: posterior encoder, prior encoder and decoder as depicted in Figure 1. The posterior encoder $\mathcal{E}(\cdot)$ composed of non-causal WaveNet [24] residual blocks models the posterior distribution $p(z|y, e)$ of the hidden representation $z = \mathcal{E}(y, e)$ from the linear spectrograms y generated from the original singing waveforms where singer embedding e is extracted by an additional speaker verification model. The prior encoder is implemented using a multi-layer Transformer [25]. Given the PPG and fundamental frequency (F0) denoted as x and f_0 respectively, the prior encoder estimates the prior distribution $p(z|x, f_0, e)$ with the target singer’s timbre and the flow. To bridge the distribution between the prior encoder and posterior encoder, normalizing flow with speaker-normalized affine couplin (SNAC) [26] layers is exploited to perform an invertible transformation of a simple distribution into a more complex one. The BigVGAN-based decoder [27] $\mathcal{D}(\cdot)$ generates the singing waveform from the latent representation z using a neural source filter (NSF) scheme [28] with F0 to enhance voice reconstruction quality.

After training the So-VITS-SVC system, we retain only the posterior encoder and decoder. The posterior encoder compresses the linear spectrogram to generate the latent representation, used as the prediction target for the diffusion model in LDM-SVC training. During inference, the latent representation is predicted from Gaussian White Noise by the denoising process, and the waveform is generated via the decoder.

2.2. Latent Diffusion

LDM is adopted as the probabilistic models that fit the hidden distribution by denoising on data latent space from pre-



trained VAE model. We use the Denoising Diffusion Probabilistic Models (DDPM) method [29] to train the diffusion model, which consists of forward and denoising processes. Initially, **we pretrain an SVC model using the So-VITS-SVC framework and combine its posterior encoder and decoder to form a VAE** (see the right part of Figure 1). **During the LDM training, the singer’s timbre e and the linear spectrogram of the singing voice y are used as inputs to the posterior encoder $\mathcal{E}(\cdot)$, yielding the latent variable $z_0 = z = \mathcal{E}(y, e)$. In the forward process,** the original data distribution is transformed into a standard Gaussian distribution by gradually adding noise according to a fixed schedule β_1, \dots, β_T . Here, T represents the total timesteps. The transition from z_0 to z_t follows a Markov chain, where the conditional distribution $q(z_t|z_{t-1})$ is defined as a Gaussian distribution: $q(z_t|z_{t-1}) = \mathcal{N}(z_t; \sqrt{1 - \beta_t}z_{t-1}, \beta_t \mathbf{I})$.

The denoising process, parameterized by θ , acts as a denoising function, eliminating the added noise and restoring the original data structure. This denoising distribution, $p_\theta(z_{t-1}|z_t)$, is modeled as a conditional Gaussian distribution. By using this parameterized denoising process, we iteratively sample the target data z_0 from a Gaussian noise for $t = T, T-1, \dots, 1$. In each iteration, z_{t-1} is sampled according to $p_\theta(z_{t-1}|z_t)$. The LDM model receives z_t and t as inputs, complemented by conditional inputs such as the singer’s timbre e , fundamental frequency f_0 , and PPG x . To put it simply, in the denoising procedure the calculation of z_{t-1} is given by

$$z_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(z_t - \frac{1 - \alpha_t}{\sqrt{1 - \bar{\alpha}_t}} \epsilon_\theta(z_t, t, x, f_0, e) \right) + \sigma_t \epsilon, \quad (1)$$

where $\epsilon \sim \mathcal{N}(0, I)$ represents Gaussian white noise, $\alpha_t = 1 - \beta_t$, and $\bar{\alpha}_t = \prod_{s=1}^t \alpha_s$. For a more detailed derivation of (1), please refer to [29]. The configuration of the diffusion model we use aligns with that in DiffSVC. The training loss of the latent diffusion model ϵ_θ is defined as the mean squared error (MSE) in the noise space:

$$\mathcal{L}_{LDM} = \|\epsilon - \epsilon_{\theta}(z_t, t, x, f_0, e)\|_2^2. \quad (2)$$

During inference, the singer’s timbre and F0 are replaced with the target singer’s attributes, defined as e_{tar} and f_0 , while

Algorithm 1 Training procedure of LDM-SVC.

Require: Conversion model $\epsilon_{\theta}(\cdot)$; training set $D_{train} = \{(x, f_0, e, y)\}_{m=1}^M$; pretrained So-VITS-SVC posterior encoder $\mathcal{E}(\cdot)$; distortion probability p_{uncond} ; N_{iter} iterations.

- 1: **for** $i = 1, 2, \dots, N_{iter}$ **do**
- 2: Sample (x, f_0, e, y) from D_{train} ;
- 3: $z_0 = \mathcal{E}(y, e)$;
- 4: $e, f_0 \leftarrow \emptyset$ with probability p_{uncond} ;
- 5: $\epsilon \sim \mathcal{N}(0, I)$;
- 6: Sample $t \sim \text{Uniform}(\{1, \dots, T\})$;
- 7: Take gradient descent step on
 $\nabla_{\theta} \|\epsilon - \epsilon_{\theta}(\sqrt{\bar{\alpha}_t} z_0 + \sqrt{1 - \bar{\alpha}_t} \epsilon, t, x, f_0, e)\|_2^2$;
- 8: **end for**
- 9: **return** $\epsilon_{\theta}(\cdot)$;

Algorithm 2 Inference procedure of LDM-SVC.

Require: Trained conversion model $\epsilon_{\theta}(\cdot)$; source singer PPG x_{src} ; **target singer embedding** e_{tar} ; modified f_0 ; pretrained So-VITS-SVC decoder $\mathcal{D}(\cdot)$; guidance weight w .

- ```

1: Sample $z_T \sim \mathcal{N}(0, I)$;
2: for $t = T, T-1, \dots, 1$ do
3: $\epsilon_t = (1+w)\epsilon_\theta(z_t, t, x, f_0, e_{tar}) - w\epsilon_\theta(z_t, t, x, \emptyset, \emptyset)$;
4: $\epsilon \sim \mathcal{N}(0, I)$ if $t > 1$ else $\epsilon = 0$;
5: $z_{t-1} = \frac{1}{\sqrt{\alpha_t}}(z_t - \frac{1-\alpha_t}{\sqrt{1-\alpha_t}}\epsilon_t) + \sigma_t\epsilon$;
6: end for
7: return $\mathcal{D}(z_0, f_0, e_{tar})$;

```

the source singer’s PPG  $x_{src}$  is used as a condition. We sample a random Gaussian White Noise in the denoising process to obtain latent variable  $z_0$ . Consequently,  $z_0$ ,  $e_{tar}$ , and  $f_0$  are fed into the pretrained decoder  $\mathcal{D}(\cdot)$  to generate the audio waveform. The use of LDM can alleviate the inconsistency caused by training-testing mismatches in So-VITS-SVC.

### 2.3. Singer Guidance

As in Figure 2, we employ the speaker condition layer normalization (SCLN) [30] to normalize the PPG feature. Additionally, we use classifier-free guidance [22] to train the model. Specifically, we expect the model to predict  $p(z|x)$  via the score estimator  $\epsilon_{\theta}(z_t, t, x)$  obtained from unconditional diffusion, while simultaneously predicting  $p(z|x, e, f_0)$  through the score estimator  $\epsilon_{\theta}(z_t, t, x, e, f_0)$  obtained from the conditional diffusion, as summarized in Algorithm 1. During model inference, we perform two inferences. In the first inference, we input all conditions normally. In the second, we set  $e_{tar}$  and  $f_0$  to an empty set. Finally, we perform sampling using the following linear combination of the conditional and unconditional score estimates with a guidance weight  $w$  as shown in Algorithm 2:

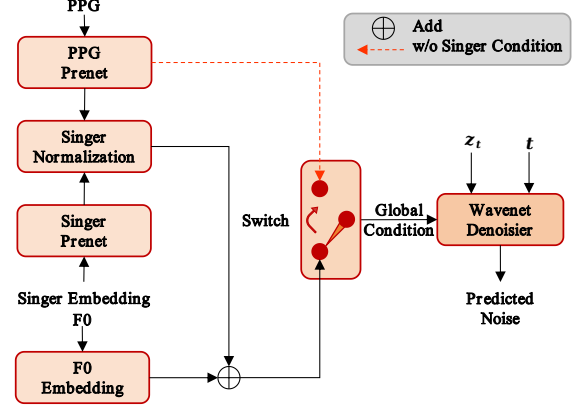
$$\epsilon_t = (1 + w)\epsilon_\theta(z_t, t, x, f_0, e_{tar}) - w\epsilon_\theta(z_t, t, x, \emptyset, \emptyset). \quad (3)$$

Based on the linear combination of predictions from both the conditional and unconditional models, we can thus more effectively reduce the timbre information of the source singer.

### 3. Experiment

### 3.1. Experiment Setup

To better accomplish the any-to-any SVC task, we choose the OpenSinger [31] dataset, which contains a large number of Chinese singers. The dataset comprises 74 singers, consisting of

Figure 2: *Singer guidance using a latent diffusion model.*

27 males and 47 females, and amounts to a total of 52 hours of recordings. To evaluate the conversion efficacy of the proposed system, we test two cases separately on whether the target singers are included in the training dataset (seen singers) or not (unseen singers). In the test set, 4 male and 4 female singers are randomly selected as the unseen singers. For the case of seen singers, 8 sentences from each remaining 66 singers are chosen for testing and the rest of songs are used for training.

We train FastSVC, DiffSVC and So-VITS-SVC models for comparison, where the So-VITS-SVC model is also the VAE used in our LDM method. To ensure a fair comparison, the PPG for each model is extracted using Whisper [32], while F0 is extracted using CREPE [33]. The singer embeddings are obtained from the pre-trained model of the CAM++ [34], a speaker verification open-source project on modelscope<sup>2</sup>. In experiments, all audio files are resampled to 32kHz. For F0 embedding in the diffusion condition, we first quantize the Log-F0 features into 256 bins and then go through a melody embedding lookup table. During inference, we re-edit the F0 of the original singer. Specifically, we calculate the mean F0 values of the target singer and source singer in the voice segment, denoted as  $\text{mean}(f_{0_v}^{src})$  and  $\text{mean}(f_{0_v}^{tar})$  respectively. Then, we multiply  $f_{0_v}^{src}$  with the ratio of these mean values to obtain the modified  $f_0$  as  $f_0^{src} \times \text{mean}(f_{0_v}^{tar}) / \text{mean}(f_{0_v}^{src})$ . For the singer guidance method, we set the distortion probability  $p_{uncond}$  during training to 0.1, and the guidance weight  $w$  during inference to 0.3 similarly to [22]. The total number of diffusion steps is 100 (i.e.,  $T = 100$ ). The noise schedule, represented as  $\beta$ , is configured to be linearly distributed, ranging from a very small value of  $1 \times 10^{-4}$  up to 0.06. This setup follows the approach used in the DiffSVC model. In our DiffSVC experiment, to ensure a fair comparison, we use BigVGAN with NSF as the unified vocoder in accordance with the structure adopted in So-VITS-SVC. Audio samples are available at <https://sounddemos.github.io/lDM-svc>.

### 3.2. Evaluation Metrics

To construct the conversion trials, we separately establish pairs of conversions for both seen and unseen scenarios and a cross-validation strategy is employed to conduct audio clips where each singer provides the vocal content information as source and other singers provide timbre information as targets.

<sup>2</sup>[https://modelscope.cn/models/iic/speech\\_campplus\\_sv\\_zh-cn\\_16k-common/summary](https://modelscope.cn/models/iic/speech_campplus_sv_zh-cn_16k-common/summary)

Table 1: Subjective indicators (SMOS, NMOS) and objective indicators (SSIM, FPC) under various method.

| Method         | Seen Singer                         |                                     |              |              | Unseen Singer                       |                                     |              |              |
|----------------|-------------------------------------|-------------------------------------|--------------|--------------|-------------------------------------|-------------------------------------|--------------|--------------|
|                | NMOS                                | SMOS                                | SSIM         | FPC          | NMOS                                | SMOS                                | SSIM         | FPC          |
| FastSVC        | $2.195 \pm 0.053$                   | $2.041 \pm 0.029$                   | 0.255        | 0.734        | $2.245 \pm 0.073$                   | $1.788 \pm 0.066$                   | 0.287        | 0.696        |
| DiffSVC        | $3.950 \pm 0.087$                   | $3.681 \pm 0.115$                   | 0.593        | 0.944        | $3.720 \pm 0.108$                   | $3.444 \pm 0.127$                   | 0.586        | 0.942        |
| So-VITS-SVC    | $3.985 \pm 0.082$                   | $3.578 \pm 0.112$                   | 0.639        | <b>0.946</b> | $3.906 \pm 0.095$                   | $3.416 \pm 0.124$                   | 0.600        | 0.942        |
| LDM-SVC-w/o-SD | $4.067 \pm 0.075$                   | <b><math>3.950 \pm 0.116</math></b> | 0.680        | 0.945        | $3.937 \pm 0.093$                   | $3.650 \pm 0.139$                   | 0.640        | 0.940        |
| LDM-SVC        | <b><math>4.125 \pm 0.080</math></b> | $3.947 \pm 0.114$                   | <b>0.702</b> | 0.945        | <b><math>4.025 \pm 0.100</math></b> | <b><math>3.841 \pm 0.127</math></b> | <b>0.663</b> | <b>0.942</b> |

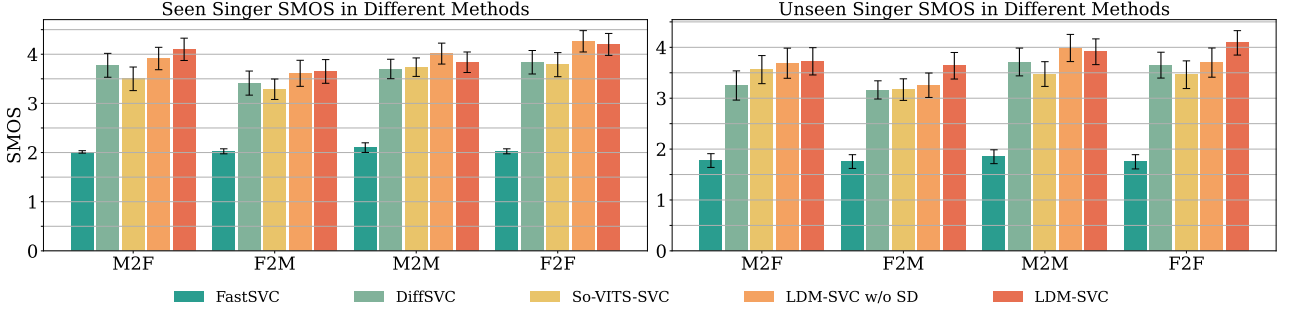


Figure 3: Detailed SMOS for seen and unseen scenarios, including M2M, M2F, F2M and F2F.

We evaluate these converted audios using F0 Pearson correlation (FPC) and Singer Similarity (SSIM), with SSIM calculated via cosine similarity using speaker embedding from the Automatic Speaker Verification model. For subjective evaluation, we randomly pick 40 audio samples from each model in both seen and unseen scenarios. These include Male-to-Male (M2M), Male-to-Female (M2F), Female-to-Male (F2M) and Female-to-Female (F2F), and each has 10 conversions. We conduct a 5-point Mean Opinion Score (1-bad, 2-poor, 3-fair, 4-good, 5-excellent) test using these randomly selected audio clips. We invite 10 volunteers to participate in the listening test, where both the Naturalness Mean Opinion Score (NMOS) and Similarity Mean Opinion Score (SMOS) are evaluated.

### 3.3. Experimental Results

Results for seen and unseen singer cases are shown in Table 1. Overall, the SMOS, NMOS, SSIM and FPC in the seen scenario are generally better than those in the unseen scenario, which is consistent with common sense. In Table 1, the fluency and similarity results of FastSVC are at a relatively low level, indicating a poor applicability to any-to-any conversion, and the generated audio is of poor quality and has no conversion effect. For the DiffSVC and So-VITS-SVC methods, the scores on NMOS are higher, but the SMOS is relatively low, implying that in case of only providing a few seconds of the target singer’s audio, these SVC models may not be able to convert source inputs to the target singer’s singing well. Comparing DiffSVC with So-VITS-SVC, it can be observed that the end-to-end generative model So-VITS-SVC produces a better sound quality than the two-stage DiffSVC model. However, the So-VITS-SVC model tends to have more severe timbre leakage. The results of proposed system LDM-SVC and the ablation system without singer guidance mechanism (LDM-SVC-w/o-SD) are also illustrated in Table 1. It is clear that both variants have higher SMOS than comparison methods and also have some improvement in

NMOS. This discrepancy may be due to the fact that we directly predict  $z$  generated by the posterior encoder, whereas during inference with the So-VITS-SVC model, the prior encoder and flow are used to predict  $z$  that is approximated by the posterior encoder during training, leading to a mismatch between training and testing. Comparing the inclusion of singer guidance, the SMOS in the seen and unseen cases are at the same level. However, in the unseen case, the similarity of the model trained with singer guidance is higher than that in the seen case, and the SMOS scores are closer. This reveals that the proposed singer guidance method is effective in the zero-shot scenario.

The SMOS results for the gender-specific SVC tests are shown in Figure 3. We see that the SMOS scores for all ‘to male’ conversions are significantly lower than those for ‘to female’ cases, which may be due to the imbalance in the dataset between male and female singers. Overall, the issue of timbre leakage in cross-gender conversion is more severe than in same-gender conversions. For example, in the case of M2M the similarity of the model using the singer guidance method is lower than that only using the LDM method, which means that singer guidance in M2M conversions is less effective than others.

## 4. Conclusion

In this paper, we proposed the LDM-SVC system, which utilizes an LDM for any-to-any singing voice conversion. We pre-trained a So-VITS-SVC model, whose posterior encoder and decoder were used to construct a VAE to handle the latent representation. It was shown that the proposed LDM-SVC method outperforms existing approaches in terms of timbre similarity in both seen and unseen singer conversion cases. The designed singer guidance method is beneficial for improving the conversion similarity under zero-shot conditions for unseen singer conversions. In the future, we will consider cross-domain SVC tasks, e.g., converting input speech into singing voice, as a necessity for low-resource scenarios without singing voice data.

## 5. References

- [1] Y. Gu, X. Yin, Y. Rao, Y. Wan, B. Tang, Y. Zhang, J. Chen, Y. Wang, and Z. Ma, "ByteSing: A Chinese singing voice synthesis system using duration allocated encoder-decoder acoustic models and wavernn vocoders," in *2021 12th International Symposium on Chinese Spoken Language Processing (ISCSLP)*. IEEE, 2021, pp. 1–5.
- [2] J. Cui, Y. Gu, C. Weng, J. Zhang, L. Chen, and L. Dai, "Sifisinger: A high-fidelity end-to-end singing voice synthesizer based on source-filter model," in *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2024, pp. 11 126–11 130.
- [3] L. Sun, K. Li, H. Wang, S. Kang, and H. Meng, "Phonetic posteriorgrams for many-to-one voice conversion without parallel data training," in *2016 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2016, pp. 1–6.
- [4] A. Polyak, L. Wolf, Y. Adi, and Y. Taigman, "Unsupervised cross-domain singing voice conversion," *arXiv preprint arXiv:2008.02830*, 2020.
- [5] 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)*. IEEE, 2021, pp. 1–6.
- [6] 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)*. IEEE, 2021, pp. 741–748.
- [7] Z. Li, B. Tang, X. Yin, Y. Wan, L. Xu, C. Shen, and Z. Ma, "Ppg-based singing voice conversion with adversarial representation learning," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 7073–7077.
- [8] T. Jayashankar, J. Wu, L. Sari, D. Kant, V. Manohar, and Q. He, "Self-supervised representations for singing voice conversion," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.
- [9] Y. Zhou, M. Chen, Y. Lei, J. Zhu, and W. Zhao, "VITS-based singing voice conversion system with dspgan post-processing for svc2023," in *2023 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2023, pp. 1–8.
- [10] E. Nachmani and L. Wolf, "Unsupervised singing voice conversion," *arXiv preprint arXiv:1904.06590*, 2019.
- [11] 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)*. IEEE, 2020, pp. 7749–7753.
- [12] L. Zhang, C. Yu, H. Lu, C. Weng, C. Zhang, Y. Wu, X. Xie, Z. Li, and D. Yu, "Durian-sc: Duration informed attention network based singing voice conversion system," *arXiv preprint arXiv:2008.03009*, 2020.
- [13] N. Takahashi, M. K. Singh, and Y. Mitsufuji, "Hierarchical disentangled representation learning for singing voice conversion," in *2021 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2021, pp. 1–7.
- [14] Y. Zhou and X. Lu, "HiFi-SVC: Fast high fidelity cross-domain singing voice conversion," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 6667–6671.
- [15] Y.-J. Luo, C.-C. Hsu, K. Agres, and D. Herremans, "Singing voice conversion with disentangled representations of singer and vocal technique using variational autoencoders," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 3277–3281.
- [16] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 10 684–10 695.
- [17] D. Ghosal, N. Majumder, A. Mehrish, and S. Poria, "Text-to-audio generation using instruction-tuned llm and latent diffusion model," *arXiv preprint arXiv:2304.13731*, 2023.
- [18] H. Liu, Z. Chen, Y. Yuan, X. Mei, X. Liu, D. Mandic, W. Wang, and M. D. Plumbley, "Audioldm: Text-to-audio generation with latent diffusion models," *arXiv preprint arXiv:2301.12503*, 2023.
- [19] H. Liu, Q. Tian, Y. Yuan, X. Liu, X. Mei, Q. Kong, Y. Wang, W. Wang, Y. Wang, and M. D. Plumbley, "Audioldm 2: Learning holistic audio generation with self-supervised pretraining," *arXiv preprint arXiv:2308.05734*, 2023.
- [20] J. Liu, C. Li, Y. Ren, F. Chen, and Z. Zhao, "Diffsinger: Singing voice synthesis via shallow diffusion mechanism," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 36, no. 10, 2022, pp. 11 020–11 028.
- [21] V. Popov, I. Vovk, V. Gogoryan, T. Sadekova, M. Kudinov, and J. Wei, "Diffusion-based voice conversion with fast maximum likelihood sampling scheme," *arXiv preprint arXiv:2109.13821*, 2021.
- [22] J. Ho and T. Salimans, "Classifier-free diffusion guidance," *arXiv preprint arXiv:2207.12598*, 2022.
- [23] J. Kim, J. Kong, and J. Son, "Conditional variational autoencoder with adversarial learning for end-to-end text-to-speech," in *International Conference on Machine Learning*. PMLR, 2021, pp. 5530–5540.
- [24] A. v. d. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," *arXiv preprint arXiv:1609.03499*, 2016.
- [25] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.
- [26] B. J. Choi, M. Jeong, J. Y. Lee, and N. S. Kim, "Snac: Speaker-normalized affine coupling layer in flow-based architecture for zero-shot multi-speaker text-to-speech," *IEEE Signal Processing Letters*, vol. 29, pp. 2502–2506, 2022.
- [27] S.-g. Lee, W. Ping, B. Ginsburg, B. Catanzaro, and S. Yoon, "Bigvgan: A universal neural vocoder with large-scale training," *arXiv preprint arXiv:2206.04658*, 2022.
- [28] X. Wang, S. Takaki, and J. Yamagishi, "Neural source-filter waveform models for statistical parametric speech synthesis," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 402–415, 2019.
- [29] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Advances in neural information processing systems*, vol. 33, pp. 6840–6851, 2020.
- [30] P. Wu, J. Pan, C. Xu, J. Zhang, L. Wu, X. Yin, and Z. Ma, "Cross-speaker emotion transfer based on speaker condition layer normalization and semi-supervised training in text-to-speech," *arXiv preprint arXiv:2110.04153*, 2021.
- [31] 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," in *Proceedings of the 29th ACM International Conference on Multimedia*, 2021, pp. 3945–3954.
- [32] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," in *International Conference on Machine Learning*. PMLR, 2023, pp. 28 492–28 518.
- [33] 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)*. IEEE, 2018, pp. 161–165.
- [34] H. Wang, S. Zheng, Y. Chen, L. Cheng, and Q. Chen, "CAM++: A fast and efficient network for speaker verification using context-aware masking," *arXiv preprint arXiv:2303.00332*, 2023.