DIFFSVC: A DIFFUSION PROBABILISTIC MODEL FOR SINGING VOICE CONVERSION

Songxiang Liu^{*1}, *Yuewen Cao*^{*1}, *Dan Su*^{2}, *Helen Meng*^{1}

^{1}Human-Computer Communications Laboratory, The Chinese University of Hong Kong
^{2}Tencent AI Lab

ABSTRACT

Singing voice conversion (SVC) is one promising technique that can enrich the way of human-computer interaction by endowing a computer the ability to produce high-fidelity and expressive singing voice. In this paper, we propose DiffSVC, an SVC system based on denoising diffusion probabilistic model. DiffSVC uses phonetic posteriorgrams (PPGs) as content features. A denoising module is trained in DiffSVC, which takes destroyed mel spectrogram produced by the diffusion/forward process and its corresponding step information as input to predict the added Gaussian noise. We use PPGs, fundamental frequency features and loudness features as auxiliary inputs to assist the denoising process. Experiments show that DiffSVC can achieve superior conversion performance in terms of naturalness and voice similarity to current state-of-the-art SVC approaches.

Index Terms— singing voice conversion, diffusion probabilistic model

1. INTRODUCTION

Singing plays an important role in human daily life, including information transmission, emotional expression and entertainment. The technology of singing voice conversion (SVC) aims at converting the voice of a singing signal to a voice of a target singer without changing the underlying content and melody. Endowing machine with the ability to produce high-fidelity and expressive singing voice provides new ways for human-computer interaction. SVC is among the possible ways to achieve this.

Most recent SVC systems train a content encoder to extract content features from a source singing signal and a conversion model to transform content features to either acoustic features or waveform. One class of SVC approaches jointly trains the content encoder and the conversion model as an auto-encoder model [1, 2]. Another class of SVC approaches separately trains the content encoder and the conversion model. These approaches train an automatic speech recognition (ASR) model as the content encoder. The ASR model can be an end-to-end model, as in [3, 4] or a hybrid HMM-DNN model, as in [5]. The conversion model can be the generator in a generative adversarial network (GAN) [3, 4], which directly generates waveform from content features; or a regression model, which transforms content features to spectral features (e.g., mel spectrograms), and adopts an additionally trained neural vocoder to generate waveform. In this paper, we focus on the latter class and devote to introducing the recently-emerged diffusion probabilistic modeling into the conversion model.

Diffusion probabilistic models, or diffusion models for brevity, are a class of promising generative models [6]. It has been demonstrated that diffusion models are capable of achieving state-of-the-art performance on generative modeling for natural images [7, 8] and raw audio waveform [9, 10]. There are two processes in a diffusion model, i.e., the diffusion/forward process and the reverse process. The diffusion/forward process is a Markov chain with fixed parameters, which converts the complicated data into isotropic Gaussian distribution by adding Gaussian noise gradually. The reverse process is also a Markov chain, which restores the data structure from Gaussian noise in an iterative manner. One merit of diffusion models is that they can be efficiently trained by optimizing the evidence lower bound (ELBO), also known as variational lower bound (VLB), of the data likelihood. In [7], one certain parameterization trick can convert the learning process of diffusion models into a regression problem. We follow this setting in this paper.

We propose DiffSVC, which is an SVC system based on diffusion model. We use an ASR acoustic model to extract phonetic posteriorgrams (PPGs) from singing signals as the content features. A diffusion model is trained to recover mel spectrograms iteratively from Gaussian noise, conditioning on content, melody and loudness features. We show that DiffSVC can achieve superior conversion performance in terms of naturalness and voice similarity, compared with current state-of-the-art SVC approaches. The contributions of our work include: (1) To the best of our knowledge, the proposed DiffSVC system is the first SVC system using the diffusion probabilistic model. We show that diffusion model can be effectively adopted for the SVC task; and (2) DiffSVC achieves better conversion performance in terms of naturalness and voice similarity than previous SVC systems.

The rest of this paper is organized as follows: Section 2 presents related work. Section 3 introduces the diffusion

[*] Equal contribution. Work done during internship in Tencent AI Lab

probabilistic model. Details of the proposed DiffSVC system are presented in Section 4. Experimental results are shown in Section 5 and Section 6 concludes this paper.

2. RELATED WORK

2.1. Singing Voice Conversion

Current SVC approaches can be categorized into two classes, i.e., parallel SVC and non-parallel SVC, in terms of whether parallel training data is required. Most initial attempts for SVC are within the parallel SVC paradigm. These approaches model parallel training samples using statistical methods, such as Gaussian mixture model (GMM)-based many-to-many eigenvoice conversion [11], direct waveform modification based on spectrum difference [12, 13]. GAN-based parallel approach has also been proposed to improve conversion performance [14]. Since parallel SVC approaches require parallel data (i.e., source and target singers sing the same contents), which is expensive to collect, for the training process, researchers have investigated many non-parallel SVC approaches. Auto-encoder based on the WaveNet [15] structure has been used for unsupervised SVC in [1], which can convert among singers appeared in the training set. This approach adopts an adversarial speaker classifier to disentangle singer information from the encoder output. To further improve this method, an additional pitch adversarial mechanism is added to remove pitch information from the encoder output in [2]. Variational auto-encoder (VAE) [16], variational autoencoding Wasserstein GAN (VAW-GAN) [17], and phonetic posteriorgram (PPG) models [5] are also investigated for non-parallel SVC. Very recently, the combination of a PPG model and a waveform generator achieves promising SVC performance [3, 4].

2.2. Diffusion Probabilistic Models

There are two streams of efforts, pushing the research on diffusion probabilistic models forward. One is research on the score matching model, which originates from [18], where the problem of data density estimation is simplified into a score matching problem (the estimation of the gradient of the data distribution density). The other is research on denoising diffusion probabilistic model, which originates from [6], where a diffusion Markov chain is used to destroy data structure into Gaussian noise and another reverse process is used to generate data from Gaussian noise. Recently, great progress has been made in these two streams. In [7], the diffusion model has been able to generate high-quality images. Slightly later, the diffusion model also makes it to generate high-quality audio samples [9, 10]. On the other hand, score matching model has also been proved to be capable of generating high-resolution images using a neural network to estimate the log gradient of the target data probabilistic density function [19]. Afterwards, a unified framework is proposed in [20], which generalizes and improves previous work on score matching model through the lens of stochastic differential equations (SDEs). The diffusion model in [7] and the score matching model in [19] have been shown to be special cases under such framework.

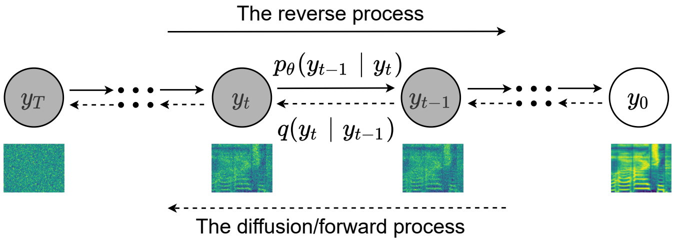

Fig. 1. Graphical model for the reverse process and the diffusion/forward process in a diffusion probabilistic model. The diffusion/forward process (dotted arrow) gradually converts data into noise within finite steps. The reverse process (solid arrow) is a parametric procedure which attempts to restore data structure from noise.

3. DIFFUSION PROBABILISTIC MODEL

In this section, we introduce the denoising diffusion probabilistic model (or diffusion model in short) [6]. We denote the data as $y_0 \sim q(y_0)$. A diffusion model is a latent variable model with the form $p_\theta(y_0) := \int p_\theta(y_{0:T}) dy_{1:T}$, where $y_1, ..., y_T$ are latent variables with the same dimensionality as the data y_0 and T is the total number of diffusion steps. As shown in Fig. 1, a diffusion model contains two processes – the diffusion/forward process and the reverse process. In the remaining part of this section, we first introduce the two processes in a diffusion model and then discuss the training and sampling algorithms.

3.1. Diffusion/Forward Process

The diffusion/forward process in a diffusion model is modeled as a Markov chain that gradually adds small noise to the data until the data structure is totally destroyed at step T. That is, the diffusion process gradually whitens the data into Gaussian noise in T diffusion steps. The chain transitions are often modeled as conditional Gaussian transitions according to a deterministic noise schedule $\beta_1, ..., \beta_T$, which are set to satisfy $\beta_1 < \beta_2 < ... < \beta_T$. The joint probability of the latent variables is:

$$q(y_{1:T}|y_0) := \prod_{t=1}^{T} q(y_t|y_{t-1}), \tag{1}$$

where $q(y_t|y_{t-1}) := \mathcal{N}(y_t; \sqrt{1-\beta_t} y_{t-1}, \beta_t I)$. The mean term $\sqrt{1-\beta_t}$ gradually decays the destroyed data towards origin, and the variance term β_t incorporates stochasticity into the process by adding small noises, for $t \in \{1, ..., T\}$.

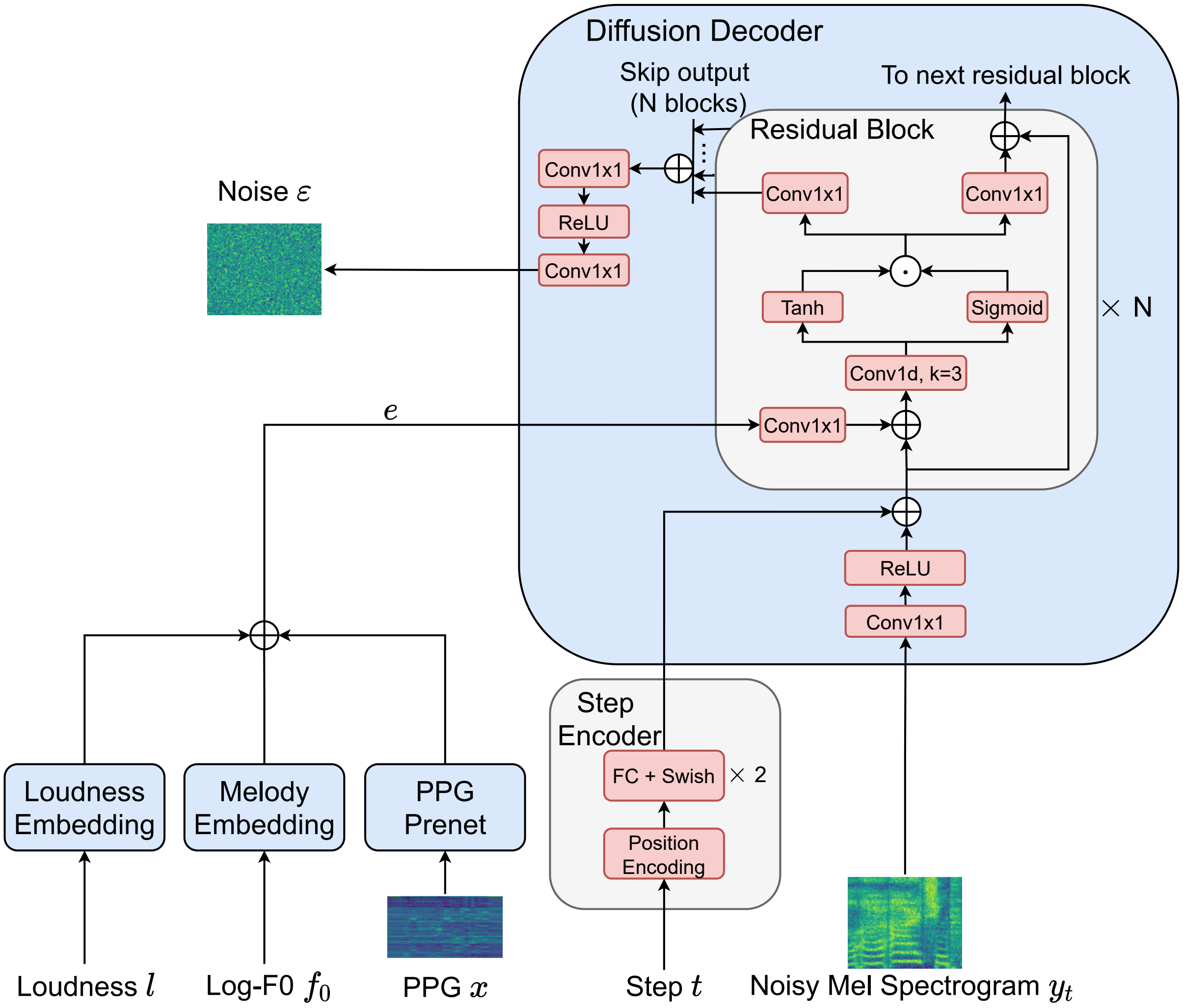

Fig. 2. Schematic diagram of the conversion model in the proposed DiffSVC system, based on the diffusion probabilistic model.

3.2. Reverse Process

The reverse process in a diffusion model is a parameterised process which gradually learns to restore data structure from white noise along another Markov chain. When β's of the diffusion/forward process are small, both the diffusion and reverse process have the same functional form [6], i.e., Gaussian transitions. Starting at isotropic Gaussian $p(y_T) = \mathcal{N}(y_T; 0, I)$, the joint distribution $p_\theta(y_{0:T})$ along the reverse process satisfies:

$$p_\theta(y_{0:T}) := p(y_T)\prod_{t=1}^{T} p_\theta(y_{t-1}|y_t), \tag{2}$$

where the transition probability $p_\theta(y_{t-1}|y_t)$ is parameterized as $\mathcal{N}(y_{t-1}; \mu_\theta(y_t,t), \sigma_\theta(y_t,t)^2 I)$ with shared parameter θ among all the T transitions. The training process of a diffusion model is to learn the parameter θ, such that $p_\theta(y_{t-1}|y_t)$ can eliminate the Gaussian noise (i.e., denoise) added during the diffusion/forward process.

3.3. Training Diffusion Model

The goal of training a diffusion model is to maximize the model likelihood $p_\theta(y_0) := \int p_\theta(y_{0:T})dy_{1:T}$, given a training set $\{y_0^m\}_{m=1}^M$ with M samples. However, the likelihood is intractable to compute in general because of the integration operation in a very high-dimensional latent space. Thus, its evidence lower bound (ELBO), which is also known as variational lower bound (VLB), is used to train the model:

$$\mathbb{E}_{q(y_0)}[\log p_\theta(y_0)] \geq \underbrace{\mathbb{E}_{q(y_{0:T})}[\log \frac{p_\theta(y_{0:T-1}|y_T)p(y_T)}{q(y_{1:T}|y_0)}]}_{:=ELBO} \tag{3}$$

We refer readers to [6] for a complete proof of Eq. (3).

Thanks to the property of Gaussian transitions in the diffusion process, the noisy data y_t (i.e., destroyed data) at any step $t \in \{1, ..., T\}$ can be readily sampled given a data sample y_0:

$$q(y_t|y_0) = \mathcal{N}(y_t; \sqrt{\bar{\alpha}_t} y_0, \sqrt{1-\bar{\alpha}_t} I), \quad (4)$$

where $\alpha_t := 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$. Following the reparameterization trick presented in [7], the learning process becomes a regression problem, where a neural network (parameterized with θ) is trained, which takes the noisy data y_t and the step variable t as input to predict the corresponding noise ϵ. Using a simplified variant of the ELBO in Eq. (3) as in [7], the negative ELBO becomes:

$$-ELBO = \mathbb{E}_{y_0,t}[||\epsilon - \epsilon_\theta(\sqrt{\bar{\alpha}_t} y_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, t)||_2^2], \quad (5)$$

where $\epsilon \sim \mathcal{N}(0, I)$, $y_0 \sim q(y_0)$, and t is uniformly sampled from $1, ..., T$. In this paper, we use Eq. (5) to train the diffusion model.

3.4. Sampling with Langevin Dynamics

Given the parameterized reverse process with the well-learned parameter θ, the sampling process (i.e., the generative process) is to first sample a Gaussian noise $y_T \sim \mathcal{N}(0, I)$, and then iteratively sample $y_{t-1} \sim p_\theta(y_{t-1}|y_t)$ for $t = T, T-1, ..., 1$ along the reverse process, according to Langevin dynamics:

$$y_{t-1} = \frac{1}{\sqrt{\alpha_t}}(y_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(y_t, t)) + \sigma_t z, \quad (6)$$

where $\sigma_t^2 = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t$, $z \sim \mathcal{N}(0, I)$ for $t > 1$ and $z = 0$ for $t = 1$. The final y_0 is the generated data.

4. DIFFSVC

In this section, we describe how we introduce the diffusion model to SVC and present details of the proposed DiffSVC system. We regard the content, melody and loudness information as the most important components in a singing signal for the task of singing voice conversion. In DiffSVC, we train a Deep-FSMN (DFSMN)-based ASR acoustic model [21] to extract PPGs as the content features with forced-aligned audio-text speech data. We train the ASR model with framewise cross-entropy loss, where the ground-truth labels are phonemes (i.e., initials and finals with tone) for Mandarin Chinese SVC.

We introduce the concept of denoising diffusion modeling into the conversion model in DiffSVC. The overall architecture is depicted in Fig. 2. During the training process, DiffSVC aims at predicting the isotropic Gaussian noise ϵ from the noisy mel spectrogram y_t and the step variable t, leveraging additional information from PPG x, logarithmic fundamental frequency feature (Log-F0) f_0 and loudness feature l. We introduce the details in the following subsections.

4.1. PPG, Log-F0 and Loudness

The PPG prenet adopts a simple fully-connected (FC) layer on the PPG input. DiffSVC represents the melody of a singing signal using its fundamental frequency contour in logarithmic scale. A-weighting mechanism of a singing signal's power spectrum is adopted to compute loudness features, following [4]. We process the Log-F0 features and loudness features in the same way. The Log-F0 features and loudness features are first quantized into 256 bins, and then go through a melody embedding lookup table and a loudness embedding lookup table respectively. The processed PPG features, Log-F0 features and loudness features are added elementwisely to obtain the conditioner e, which is token as additional input to the diffusion decoder.

4.2. Diffusion Modeling

Step Encoding. The diffusion step variable t is an important input to a diffusion model, because it indicates the amount of noise added to the noisy mel spectrogram y_t. Following the setting in [9], we first convert integral-valued t into an 128-dimensional vector t_{emb} with a sinusoidal position encoding as:

$$[\sin(10^{\frac{0\times4}{63}}t), ..., \sin(10^{\frac{63\times4}{63}}t), \cos(10^{\frac{0\times4}{63}}t), ..., \cos(10^{\frac{63\times4}{63}}t)] \quad (7)$$

Then we further process t_{emb} with two FC layers and Swish activation function [22].
Diffusion Decoder. The diffusion decoder is trained to predict the noise ϵ added by the diffusion process from the noisy mel spectrogram y_t and the step variable t, taking auxiliary conditioners e obtained from PPG x, Log-F0 f_0 and loudness l. Following [9], the diffusion decoder in DiffSVC adopts a bidirectional residual convolutional architecture proposed in [15] with a slight difference. We use dilation rate of 1 since we work on mel spectrograms instead of raw waveform. The receptive field of the diffusion decoder is sufficiently large with a dilation rate of 1. The diffusion decoder first conducts an one-dimensional convolutional operation with kernel-size 1 (Conv1x1) on the noisy mel spectrogram and then adopts the ReLU activation on the output. The output of the step encoder is added to every time step of the mel spectrogram feature maps, and then fed into N residual blocks. The conditioner e is imported into every residual block with separate Conv1x1 layers, whose output is elementwisely added with the mel spectrogram and step features. Then the gated mechanism introduced in [15] are used to further process the feature maps. We sum the skip connections from all the N residual layers and further process with two Conv1x1 layers interleaved with a ReLU activation to get the diffusion decoder output.
Training and conversion We summarize the training procedure of the conversion model in DiffSVC in Algorithm 1 and the conversion procedure in Algorithm 2, respectively.

Algorithm 1 Training procedure of the conversion model in DiffSVC.

```
Require: The conversion model $\epsilon_\theta(\cdot)$; the training set
    $\mathcal{D}_{train} = \{(x, f_0, l, y_0)\}_{m=1}^{M}$; $N_{iter}$ training iterations.
for $i = 1, 2, ..., N_{iter}$ do
    Sample $(x, f_0, l, y_0)$ from $\mathcal{D}_{train}$;
    $\epsilon \sim \mathcal{N}(0, I)$;
    Sample $t \sim$ Uniform($\{1, \cdots, T\}$);
    Take gradient descent step on
        $\nabla_\theta ||\epsilon - \epsilon_\theta(\sqrt{\bar{\alpha}_t} y_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon, t, x, f_0, l))||_2^2$;
end for
return $\epsilon_\theta(\cdot)$;
```

Algorithm 2 Conversion procedure of DiffSVC.

```
Require: The trained conversion model $\epsilon_\theta(\cdot)$; one testing
    sample $(x, f_0, l)$.
Sample $y_T \sim \mathcal{N}(0, I)$;
for $t = T, T - 1, ..., 1$ do
    if $t > 1$ then
        Sample $z \sim \mathcal{N}(0, I)$;
    else
        $z = 0$;
    end if
    $y_{t-1} = \frac{1}{\sqrt{\alpha_t}}(y_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(y_t, t, x, f_0, l)) + \sigma_t z$, as in
    Eq. 6;
end for
return $y_0$;
```

5. EXPERIMENTS

We evaluate the proposed DiffSVC by conducting any-to-one SVC tasks, where the singing voice of an arbitrary source singer is converted to the target singer's. In this section, we first introduce the corpus for the experiments and the data preprocessing methods. Then the implementation details of both the baseline approaches and proposed approach are presented. Finally, we present the evaluation results.

5.1. Dataset and preprocessing

The PPG extractor is trained using an internal Mandarin Chinese ASR corpus, which contains around 20k hours speech data recorded by thousands of speakers. The PPG features have dimensionality of 218. The conversion model in DiffSVC is trained with an internal dataset, which contains 14 hours singing data recorded by a female voice talent. The audio format is 16-bit PCM with a sample rate of 24 kHz. Mel spectrograms have 80 frequency bins, and are computed with 1024-point fast Fourier transform (FFT), 1024-point window size and 240-point hop size (i.e., 10 ms). Mel spectrograms are min-max normalized to the range [-1, 1]. Fundamental frequency (F0) values are computed with hop size of 240. To make the F0 computation robust, we use three F0 estimators, i.e., DIO [23], REAPER [1] and SPTK [24]. The F0 values take the median results by this three F0 estimators. The FFT size, window size and hop size for computing loudness features are 2048, 2048 and 240, respectively.

5.2. Implementation details

The output size of the PPG prenet, the melody embedding size and loudness embedding size are all 256. The number of bins for quantizing Log-F0 and loudness is 256. The step encoder has the same hyper-parameter setting as in [9]. We use 20 residual layers in a diffusion decoder, where the number of convolution channels are 256. The total number of diffusion steps is 100 (i.e., T = 100). The noise schedule β's are set to linearly spaced from 1×10^{-4} to 0.06. We train models with the ADAM optimizer [25] with a constant learning rate of 0.0002.

We train a Hifi-GAN model [26] as vocoder to convert mel spectrograms to the raw waveform. The official implementation[2] (the "config_v1.json" configuration) is used with a slight change. Since we use 24 kHz audio samples and the hop-size for computing mel spectrogram is 240, we factorize the upsample rate as $240 = 8 \times 5 \times 3 \times 2$. To avoid possible checkerboard artifacts caused by the "ConvTranspose1d" upsampling layer, we use temporal nearest interpolation layer followed by a one-dimensional convolutional layer as the upsampling operation, as suggested in [27]. The kernel sizes we use are [15,15,5,5].

5.3. Comparisons

We compare the proposed DiffSVC system with three state-of-the-art SVC approaches. To make the comparison fair, all models use the same DFSMN-based PPG extractor (as introduced in Section 4) to compute PPG features as the content representations. Fundamental frequency (f_0) values are computed using the robust ensemble method introduced in Section 5.1. The details of their implementations are presented as follows.

BLSTM-SVC This model is proposed by [28]. We make slight modifications to make the comparison fair: (1) We use mel spectrogram and Log-F0 as the acoustic features instead of the conventional vocoder parameters such as mel cepstral (MCEP) features and aperiodicity (AP) features; and (2) we use the Hifi-GAN vocoder introduced in Section 5.2 for high-fidelity waveform generation. We reimplemented the 512N model in [28] since it shows better conversion performance according to the experimental results in the orginal paper.

Seq2seq-SVC This model is proposed by [5], where a sequence-to-sequence (seq2seq) based model using GMM-based attention mechanism is trained to predict mel spectrograms from PPGs. Slight modifications are also made to

[1] [Online] `https://github.com/google/REAPER`

[2] [Online] `https://github.com/jik876/hifi-gan`

make the comparison fair: (1) we do not use the Mel encoder and the singer confusion module because we focus on single-speaker conversion model; (2) we use the Hifi-GAN vocoder introduced in Section 5.2 for converting mel spectrograms into waveform, instead of using a WaveRNN vocoder[29].

FastSVC This model is presented in [4], which uses a generator in a GAN model to convert content features directly into waveform, and is trained by combining a multi-scale spectral loss and an adversarial loss. In the original paper, the content features used by the authors have a hop-size of 320, while the PPGs used in this paper have a hop-size of 240. Therefore, we change the upsampling rates of the generator to [5,4,4,3] from the original [4,4,4,5]. To further adapt FastSVC for 24 kHz audio synthesizing (originally, FastSVC works on 16 kHz audio), we add the multi-period discriminator (MPD) [26] as additional discriminator. These changes make FastSVC achieving better conversion performance on 24 kHz audio samples.

5.4. Evaluations

Subjective evaluations in terms of both the naturalness and voice similarity of converted speech are conducted. We use the 5 point Likert scale for testing with mean opinion score (MOS) (1-bad, 2-poor, 3-fair, 4-good, 5-excellent). In the MOS tests for evaluating naturalness, each group of stimuli contains recording samples from the target singer, which are randomly shuffled with the samples generated by the four comparative approaches, before they are presented to raters. In the MOS similarity tests, converted speech samples are directly compared with the recording samples of the target singer. 10 utterances from an evaluation set are presented for each system. We invited raters to participate in the evaluations in a quiet room and they were asked to use headphones during the tests. The raters were allowed to replay each sample as many times as necessary and change their ratings of any sample before submitting their results. The MOS results are demonstrated in Table 1. We can see that DiffSVC achieves significantly better performance than the other three compared systems (i.e., BLSTM-SVC, Seq2seq-SVC and FastSVC), in terms of both naturalness and voice similarity. Audio samples can be found online[3].

We use mel-cepstrum distortion (MCD) and F0 Pearson correlation (FPC) as metrics for objective evaluations. We extract the mel-cepstrum features of the generated audio samples and the ground-truth audio samples from the target singer for the MCD evaluation. For the FPC evaluation, we extract the F0 features from the generated audio samples and the recordings from the source singer. This is because we aim to convert the voice of a singing signal to the target singer's voice without changing the underlying content and melody. The DTW algorithm is adopted to align the predicted acoustic features towards the natural ones. The objective results are presented in Table 1. We can see that, among the four systems, DiffSVC achieves the best MCD result (6.307). In terms of FPC, the FastSVC system achieves the best result (0.904). DiffSVC achieves slightly lower FPC result than FastSVC, with an FPC value of 0.902 and an 0.2% relative drop.

Table 1. Subjective and objective evaluation results. "Nat." is for "Naturalness" and "Sim." is for "Voice Similarity", repectively. Up arrows mean "higher the better". Down arrows mean "lower the better".

System	Subjective		Objective	
	Nat. (↑)	Sim.(↑)	MCD (↓)	FPC(↑)
Recordings	4.60±0.09	-	-	-
BLSTM-SVC	2.37±0.15	3.67±0.15	6.424	0.781
Seq2seq-SVC	2.77±0.17	3.57±0.15	7.175	0.885
FastSVC	3.83±0.13	4.17±0.13	6.422	**0.904**
DiffSVC	**3.97±0.14**	**4.67±0.10**	**6.307**	0.902

Combining the experimental results, we can safely say that DiffSVC achieves superior singing voice conversion performance than previous SVC approaches in terms of naturalness and voice similarity of the converted signal.

6. CONCLUSION

In this paper, we have presented DiffSVC. To the best of our knowledge, DiffSVC is the first SVC approach using a diffusion probabilistic model to generate acoustic features (e.g., mel spectrogram) from content features (e.g., PPG). Experiments demonstrate that the diffusion model can be effectively adopted for the SVC task. Moreover, DiffSVC can achieve superior conversion performance in terms of naturalness and voice similarity than current state-of-the-art SVC systems, according to the subjective and objective evaluations. In the work, we focus on any-to-one SVC task. Future work includes further extending DiffSVC for any-to-many/any SVC. Investigation on training a DiffSVC model under low-data-resource scenario is also deserved research effort.

7. ACKNOWLEDGEMENTS

This project is partially supported by a grant from the HKSAR Government's Research Grants Council General Research Fund (Project no. 14208817).

8. REFERENCES

[1] Eliya Nachmani and Lior Wolf, "Unsupervised singing voice conversion," *INTERSPEECH*, 2019.

[2] Chengqi Deng, Chengzhu Yu, Heng Lu, Chao Weng, and Dong Yu, "Pitchnet: Unsupervised singing voice

[3] [Online] https://liusongxiang.github.io/diffsvc/

conversion with pitch adversarial network," in *ICASSP*. IEEE, 2020, pp. 7749–7753.

[3] Adam Polyak, Lior Wolf, Yossi Adi, and Yaniv Taigman, "Unsupervised cross-domain singing voice conversion," *INTERSPEECH*, 2020.

[4] Songxiang Liu, Yuewen Cao, Na Hu, Dan Su, and Helen Meng, "Fastsvc: Fast cross-domain singing voice conversion with feature-wise linear modulation," *ICME*, 2021.

[5] Zhonghao Li, Benlai Tang, Xiang Yin, Yuan Wan, Ling Xu, Chen Shen, and Zejun Ma, "Ppg-based singing voice conversion with adversarial representation learning," in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 7073–7077.

[6] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli, "Deep unsupervised learning using nonequilibrium thermodynamics," in *International Conference on Machine Learning*. PMLR, 2015, pp. 2256–2265.

[7] Jonathan Ho, Ajay Jain, and Pieter Abbeel, "Denoising diffusion probabilistic models," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, Eds. 2020, vol. 33, pp. 6840–6851, Curran Associates, Inc.

[8] Jiaming Song, Chenlin Meng, and Stefano Ermon, "Denoising diffusion implicit models," in *International Conference on Learning Representations*, 2021.

[9] Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro, "Diffwave: A versatile diffusion model for audio synthesis," *arXiv preprint arXiv:2009.09761*, 2020.

[10] Nanxin Chen, Yu Zhang, Heiga Zen, Ron J Weiss, Mohammad Norouzi, and William Chan, "Wavegrad: Estimating gradients for waveform generation," *arXiv preprint arXiv:2009.00713*, 2020.

[11] Tomoki Toda, Yamato Ohtani, and Kiyohiro Shikano, "One-to-many and many-to-one voice conversion based on eigenvoices," in *ICASSP*. IEEE, 2007, vol. 4, pp. IV–1249.

[12] Kazuhiro Kobayashi, Tomoki Toda, Graham Neubig, Sakriani Sakti, and Satoshi Nakamura, "Statistical singing voice conversion with direct waveform modification based on the spectrum differential," in *INTERSPEECH*, 2014.

[13] Kazuhiro Kobayashi, Tomoki Toda, Graham Neubig, Sakriani Sakti, and Satoshi Nakamura, "Statistical singing voice conversion based on direct waveform modification with global variance," in *INTERSPEECH*, 2015.

[14] Berrak Sisman, Karthika Vijayan, Minghui Dong, and Haizhou Li, "Singan: Singing voice conversion with generative adversarial networks," in *APSIPA ASC)*. IEEE, 2019, pp. 112–118.

[15] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu, "Wavenet: A generative model for raw audio," *ISCA SSW*, 2016.

[16] Yin-Jyun Luo, Chin-Cheng Hsu, Kat Agres, and Dorien Herremans, "Singing voice conversion with disentangled representations of singer and vocal technique using variational autoencoders," in *ICASSP*. IEEE, 2020, pp. 3277–3281.

[17] Junchen Lu, Kun Zhou, Berrak Sisman, and Haizhou Li, "Vaw-gan for singing voice conversion with non-parallel training data," in *2020 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, 2020, pp. 514–519.

[18] Aapo Hyvärinen, "Estimation of non-normalized statistical models by score matching," *Journal of Machine Learning Research*, vol. 6, no. 24, pp. 695–709, 2005.

[19] Yang Song and Stefano Ermon, "Generative modeling by estimating gradients of the data distribution," in *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds. 2019, vol. 32, Curran Associates, Inc.

[20] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole, "Score-based generative modeling through stochastic differential equations," in *International Conference on Learning Representations*, 2021.

[21] Shiliang Zhang, Ming Lei, Zhijie Yan, and Lirong Dai, "Deep-fsmn for large vocabulary continuous speech recognition," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5869–5873.

[22] Prajit Ramachandran, Barret Zoph, and Quoc V Le, "Searching for activation functions," *arXiv preprint arXiv:1710.05941*, 2017.

[23] Masanori Morise, Hideki Kawahara, and Haruhiro Katayose, "Fast and reliable f0 estimation method based on the period extraction of vocal fold vibration

of singing voice and speech," in *Audio Engineering Society Conference: 35th International Conference: Audio for Games*. Audio Engineering Society, 2009.

[24] Keiichi Tokuda, Keiichiro Oura, and et al., "Speech signal processing toolkit (sptk)," in *http://sp-tk.sourceforge.net*, 2009.

[25] Diederik P Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *ICLR*, 2015.

[26] Jungil Kong, Jaehyeon Kim, and Jaekyoung Bae, "Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, Eds. 2020, vol. 33, pp. 17022–17033, Curran Associates, Inc.

[27] Augustus Odena, Vincent Dumoulin, and Chris Olah, "Deconvolution and checkerboard artifacts," *Distill*, 2016.

[28] Xin Chen, Wei Chu, Jinxi Guo, and Ning Xu, "Singing voice conversion with non-parallel data," in *2019 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*. IEEE, 2019, pp. 292–296.

[29] Nal Kalchbrenner, Erich Elsen, Karen Simonyan, Seb Noury, Norman Casagrande, Edward Lockhart, Florian Stimberg, Aaron Oord, Sander Dieleman, and Koray Kavukcuoglu, "Efficient neural audio synthesis," in *International Conference on Machine Learning*. PMLR, 2018, pp. 2410–2419.