REALSINGER: ULTRA-REALISTIC SINGING VOICE GENERATION VIA STOCHASTIC DIFFERENTIAL EQUATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Synthesizing high-quality singing voice from music score is a challenging problem in music generation and has many practical applications. Samples generated by existing singing voice synthesis (SVS) systems can roughly reflect the lyrics, pitch and duration in a given score, but they fail to contain necessary details. In this paper, based on stochastic differential equations (SDE) we propose RealSinger to generate 22.05kHz ultra-realistic singing voice conditioned on a music score. Our RealSinger learns to find the stochastic process path from a source of white noise to the target singing voice manifold under the conditional music score, allowing to sing the music score while maintaining the local voice details of the target singer. During training, our model learns to accurately predict the direction of movement in the ambient Euclidean space onto the low-dimensional singing voice manifold. RealSinger's framework is very flexible. It can either generate intermediate feature representations of the singing voice, such as mel-spectrogram, or directly generate the final waveform, as in the end-to-end style which rectify defects and accumulation errors introduced by two-stage connected singing synthesis systems. An extensive subjective and objective test on benchmark datasets shows significant gains in perceptual quality using RealSinger. The mean opinion scores (MOS) obtained with RealSinger are closer to those of the human singer's original high-fidelity singing voice than to those obtained with any state-of-the-art method. Audio samples are available at https://realsinger.github.io/.

1 INTRODUCTION

Synthesizing ultra-realistic singing voices from music score (lyrics, notes and duration) is an important problem and has tremendous applications, including artificial intelligence singer, music-editing, and computer-aided composing. Singing voice synthesis (SVS) technology consists of two important and relatively independent modules, one is to convert music score into acoustic voice features, such as mel-spectrogram and the other is vocoder, which transforms voice features into singing waveform.

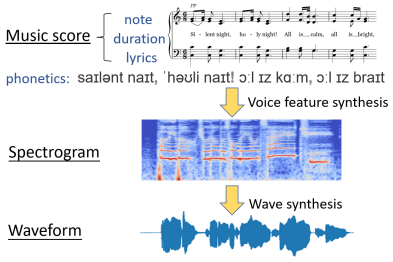

Figure 1: SVS system consists of two components, score-to-feature and vocoder.

Recently the state-of-the-art (SOTA) SVS systems are based on deep learning. Many widely studied voice generative models, such as generative adversarial network (GAN) (Goodfellow et al., 2014; Chen et al., 2021; Huang et al., 2021), Tacotron (Wang et al., 2017; 2022b), FastSpeech (Ren et al., 2020a; Wang et al., 2022a; Dong et al., 2022), etc., can be used for this task. But previous work study score-to-feature (Liu et al., 2021) and vocoder (Huang et al., 2021) separately, and design different algorithms for both tasks. In this paper we propose a flexible framework called RealSinger, which is based on Itô stochastic differential equations (SDE) for both of the components in SVS.

The generative model based on SDE (Song et al., 2020) is a kind of diffusion probability model, which has recently shown promising results in the field of image (Ho et al., 2020) and audio synthesis (Kong et al., 2020; Popov et al., 2021), even surpassing other SOTA GAN models (Dhariwal & Nichol, 2021). Prior generative models, such as GAN, flow-based model (Valle et al., 2020), or variational auto-encoder (VAE) based model (Liu et al., 2021) can be understood as a direct data distribution transformation approach, which use a crafted neural network and special training methods to achieve the transfer between white noise and the target image or audio distribution. In contrast to these previous methods, the diffusion probability model decomposes this data transformation into thousands of steps, and each of which is a simple Gaussian sampling. If the number of steps in the diffusion model tends to infinity, it becomes an SDE. The generation method based on SDE is very flexible, we can design different drift and diffusion coefficients to achieve infinitely accurate transformation between data distributions, and also this method has no special structure requirements for neural networks. The purpose of this paper is to explore SDE to achieve ultra-realistic singing voice generation.

Specifically, we make the following contributions:

- To the best of our knowledge, RealSinger is the first SVS system based on solving Itô SDEs.
- RealSinger is flexible that it can generate both middle representation feature (e.g. mel-spectrogram) and waveform (end-to-end) of singing voice
- RealSinger can generate ultra-realistic singing voices that are comparable to human being's.

2 BACKGROUND ON SDE BASED GENERATIVE MODELLING

Itô linear SDE is an excellent and tractable model for converting between different probability distributions (Song et al., 2021). The general form of Itô linear SDE is as follows

$$\begin{cases} d\mathbf{X} = [\mathbf{C}(t)\mathbf{X} + \mathbf{d}(t)]\, dt + g(t)\mathbf{I}d\mathbf{W} \\ \mathbf{X}(0) = \mathbf{x}(0). \end{cases} \tag{1}$$

for $0 \leq t \leq T$, where $\mathbf{x}(t) \in \mathbb{R}^d$, $\mathbf{C}(t) \in \mathbb{R}^{d\times d}$, $\mathbf{d}(t) \in \mathbb{R}^d$, $[\mathbf{C}(t)\mathbf{X} + \mathbf{d}(t)]$ is the drift coefficient, $g(t)$ is the diffusion coefficient, $\mathbf{W}$ is the standard Wiener process. Let $p(\mathbf{x}(t))$ be the probability density of the stochastic variable $\mathbf{X}(t)$. This SDE (1) transforms the beginning distribution $p(\mathbf{x}(0))$ into the final distribution $p(\mathbf{x}(T))$ by gradually adding the noise from $\mathbf{W}$. In this paper, $p(\mathbf{x}(0))$ is to denote the probability distribution of singing voice. $p(\mathbf{x}(T))$ is the Gaussian latent representation of the singing voice corresponding to the music score.

If the solution process $\mathbf{x}(t)$ can be reversed in time, the target singing voice corresponding to the musical score can be generated from a simple latent distribution.

Indeed the reverse-time process is the solution of the corresponding reverse-time SDE (Anderson, 1982)

$$\begin{cases} d\mathbf{X} = \left[\mathbf{C}(t)\mathbf{X} + \mathbf{d}(t) - g(t)^2\nabla_{\mathbf{x}} \log p(\mathbf{x}(t))\right] dt + g(t)d\overline{\mathbf{W}} \\ \mathbf{X}(T) = \mathbf{x}(T) \end{cases} \tag{2}$$

for $0 \leq t \leq T$, where $\overline{\mathbf{W}}$ is the standard Wiener process in reverse-time. Therefore, it can be seen that the key to generating singing voice with SDE lies in the calculations of $\nabla_{\mathbf{x}} \log p(\mathbf{x}(t))$ $(0 \leq t \leq T)$, which is always called score[1] function (Hyvärinen & Dayan, 2005; Song et al., 2021) of the singing voice. And the score function can be obtained by optimizing the denoising score matching loss (Hyvärinen & Dayan, 2005; Vincent, 2011).

3 REALSINGER

RealSinger is based on Itô SDE and score matching in principle. Conditioned on the input musical score, the Itô SDE with mel (or wave) score as its drift coefficient can continuously transform targe

[1]There are three 'score's in this paper, the first is the musical score, which represents the combination of lyrics, notes and duration; the second is the gradient of the log value of the probability distribution; the last is the mean opinion score, which is a subjective scalar measure of speech quality. Readers can easily tell the difference based on the context.

mel-spectrogram (or raw wave) into Gaussian noise by gradually injecting small-scaled white noise into it. Then the corresponding reverse Itô SDE can be used to generate the target voice from random Gaussian noise, under the conditional text-speaker joint (or mel-spectrogram) input.

To overcome the problem of deviating from the expected path during the diffusion process, DiffSinger (Liu et al., 2021) diffuses the target acoustic features predicted by the FFT model for a very few steps, and then do the reverse diffusion process to obtain the target acoustic features. To achieve ultra-realistic voices, RealSinger takes a completely different approach. We first replace discrete DDPM (Ho et al., 2020) with a continuous SDE diffusion model, which can be much more precise and flexible in training and sampling with numerical differential equation solvers. The second is to add the information of the target singing voice to the drift coefficient, so that the solution process will not deviate from the expected path in the process of reverse diffusion. RealSinger is based on the following generalized linear variance preserving (VP) SDE (Song et al., 2021)

$$\begin{cases} d\mathbf{X} = -\frac{1}{2}\beta(t)(\mathbf{X} - M)dt + \sqrt{\beta(t)}d\mathbf{W}, \quad t \in [0, 1] \\ \mathbf{X}(0) = \mathbf{x}(0) \sim p_{mel}(\cdot|MS), \end{cases} \tag{3}$$

where $\beta(t) = \beta_0 + t(\beta_1 - \beta_0)$, M is a constant that contains the information of current expected target singing voice, and MS is the conditional input music scores, which include lyrics, duration and pitch. The VP SDE (3) is a special case of linear SDE (1).

3.1 Score function and approximation

For general SDE, its score function is very difficult to calculate. Luckily, for linear SDE, we can get the score by solving a deterministic differential equation. The transition densities $p(\mathbf{x}(t)|\mathbf{x}(0))$ of the solution process $\mathbf{X}(t)$ for the SDE (3) is the solution to the Fokker-Planck-Kolmogorov (FPK) equation (Särkkä & Solin, 2019)

$$\frac{\partial p(\mathbf{x}(t), t)}{\partial t} = -\sum_{i=1}^{d} \frac{\partial \left[-\frac{1}{2}\beta(t)(\mathbf{x} - M)\right]}{\partial x_i} + \sum_{i=1}^{d}\sum_{j=1}^{d} \frac{\partial^2}{\partial x_i \partial x_j}[\beta(t)p(\mathbf{x}(t), t)], \tag{4}$$

which in this case can derive that $p(\mathbf{x}(t)|\mathbf{x}(0))$ is Gaussian with mean $\mathbf{m}(t)$ and variance $\mathbf{V}(t)$ satisfy the ordinary equations (Särkkä & Solin, 2019)

$$\begin{cases} \frac{d\mathbf{m}(t)}{dt} = -\frac{1}{2}\beta(t)(\mathbf{m}(t) - M) \\ \frac{d\mathbf{V}(t)}{dt} = -\beta(t)\mathbf{V}(t) + \beta(t)\mathbf{I}. \end{cases}$$

By solving the above linear ordinary differential equations with initial conditions of $\mathbf{m}(0) = \mathbf{x}(0)$ and $\mathbf{V}(0) = 0$, we obtain

$$\begin{cases} \mathbf{m}(t) = \exp\left\{\left[-\frac{1}{2}B(t)\right]\right\}\mathbf{x}(0) + (I - \exp\left\{\left[-\frac{1}{2}B(t)\right]\right\})M \\ \mathbf{V}(t) = \mathbf{I} - \mathbf{I}\exp\{[-B(t)]\}, \end{cases} \tag{5}$$

where $B(t) = \int_0^t \beta(s)ds = \beta_0 t + \frac{1}{2}t^2(\beta_1 - \beta_0)$.

Therefore the score of the SDE (3) is

$$\nabla_{\mathbf{x}(t)} \log p(\mathbf{x}(t)|\mathbf{x}(0), MS) = -\mathbf{V}(t)^{-1}\left[\mathbf{x}(t) - \mathbf{m}(t)\right]. \tag{6}$$

and prior distribution $p(\mathbf{x}(T))$ is a Gaussian

$$\mathcal{N}\left(\mathbf{x}(T); M, \mathbf{I}\right) = \frac{\exp\left(-\frac{1}{2}\left\|\mathbf{x}(T) - M\right\|^2\right)}{\sqrt{(2\pi)^d}}, \tag{7}$$

which shows that the sampling at the beginning is in the vicinity of M, so that the reverse diffusion process can be controlled not to deviate from the expected path.

It can be seen from Eq. (5) and 6 that $\mathbf{m}(t)$ is dependent on $\mathbf{x}(0)$, so the score in Eq. 6 is unknown. A neural network $\mathfrak{S}_\theta$ called denoiser is used to approximate the score function, where θ denotes the parameters of the network. The input of the network $\mathfrak{S}_\theta$ includes time t, $\mathbf{x}(t)$, an approximated mel-spectrogram M (which is the same as the M in the SDE (3)), conditional input musical score

MS. The expected output is $\nabla_{\mathbf{x}(t)} \log p(\mathbf{x}(t)|\mathbf{x}(0), MS)$. The following denoising score matching (DSM) loss (Vincent, 2011; Song & Ermon, 2019)

$$\text{DSM loss} = \mathbb{E}_{t\sim[0,T]}\mathbb{E}_{\mathbf{x}(0)\sim p_{mel}(\mathbf{x}(0))}\mathbb{E}_{\mathbf{x}(t)\sim p(\mathbf{x}(t)|\mathbf{x}(0))} \left[\frac{1}{2} \parallel \mathfrak{S}_\theta(\mathbf{x}(t), t, M, MS) - \nabla_{\mathbf{x}(t)} \log p(\mathbf{x}(t)|\mathbf{x}(0)) \parallel^2\right] \tag{8}$$

is used in this paper to train the score prediction network. After we obtain $\nabla_{\mathbf{x}(t)} \log p(\mathbf{x}(t)|\mathbf{x}(0), MS)$, then we can sample $\mathbf{x}(T)$ from Eq. (7) and use the following reverse-time linear VP SDE

$$\begin{cases} d\mathbf{X} = \left[-\frac{1}{2}\beta(t)(\mathbf{X} - M) - \beta(t)\nabla_{\mathbf{x}} \log p(\mathbf{x}(t))\right] dt + \sqrt{\beta(t)}d\overline{\mathbf{W}} \\ \mathbf{X}(T) = \mathbf{x}(T) \end{cases} \tag{9}$$

to generate the singing voice corresponding to the music score.

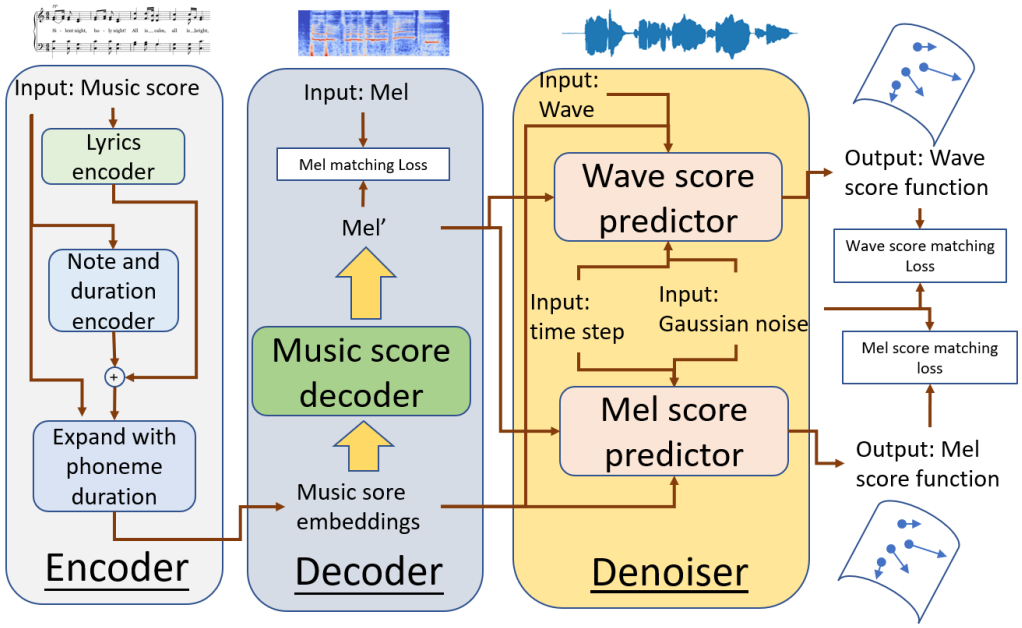

Figure 2: The training pipeline of RealSinger.

3.2 Training

RealSinger adopts an encoder-decoder-denoiser structure as shown in the Figure 2. The encoder is to encode and add the lyrics, notes and duration information in the music score together, and then expand according to the duration to obtain the final musical score embedding MS, which will be the input to the decoder and denoiser. The decoder will transcribe the music score embedding MS into an approximate corresponding singing voice spectrogram M'. The approximate spectrogram has two usages, one is as a benchmark in the diffusion SDE (3), and the other is as a conditional input in the score prediction. The denoiser is used to approximate the mel-spectrogram M' and music score embedding to predict the score function of the current signal $\mathbf{x}(t)$ at the time t.

In order to make the approximated mel M' obtained by the decoder as close as possible to the ground truth M, a $\mathcal{L}2$ loss $\parallel M - M' \parallel^2$ is used. So as to train the denoiser, we need to push the output of $\mathfrak{S}_\theta(\mathbf{x}(t), t, M', MS)$ to be the same as $\nabla_{\mathbf{x}(t)} \log p(\mathbf{x}(t)|\mathbf{x}(0), MS)$ in Eq. (6). Thus we obtain the training procedure of the score network $\mathfrak{S}_\theta$, as shown in Algorithm 1.

3.3 Sing the score

After we get the optimal encoder-decoder-denoiser through Algorithm 1, we can get the gradient of the log probability density of the mel-spectrogram of certain music score with

Algorithm 1 Training of RealSinger.

Input and initialization: The mel-spectrogram M with the corresponding music score, and the diffusion time T.
1: **for** $k = 0, 1, \cdots$
2: Uniformly sample t from $[0, T]$.
3: Randomly sample batch of M with corresponding music score, let $\mathbf{x}(0) = M$.
4: Use the encoder and decoder to obtain the music score embedding MS and the approximated mel-spectrogram M' respectively.
5: Sample $\mathbf{x}(t)$ from the Eq. (5) and Eq. (6).
6: Compute $\nabla_{\mathbf{x}(t)} \log p(\mathbf{x}(t)|\mathbf{x}(0), MS)$ with Eq. (6).
7: Use the denoiser to obtain the score $\mathfrak{S}_\theta(\mathbf{x}(t), t, M', MS)$.
8: Average the sum of the DSM loss (8) and the $\mathcal{L}2$ loss $\| M - M' \|^2$.
9: Do the back-propagation and the parameter updating of $\mathfrak{S}_\theta$.
10: $k \leftarrow k + 1$.
11: **Until** stopping conditions are satisfied and $\mathfrak{S}_{\theta_k}$ converges, e.g. to $\mathfrak{S}_{\theta_*}$.
Output: $\mathfrak{S}_{\theta_*}$.

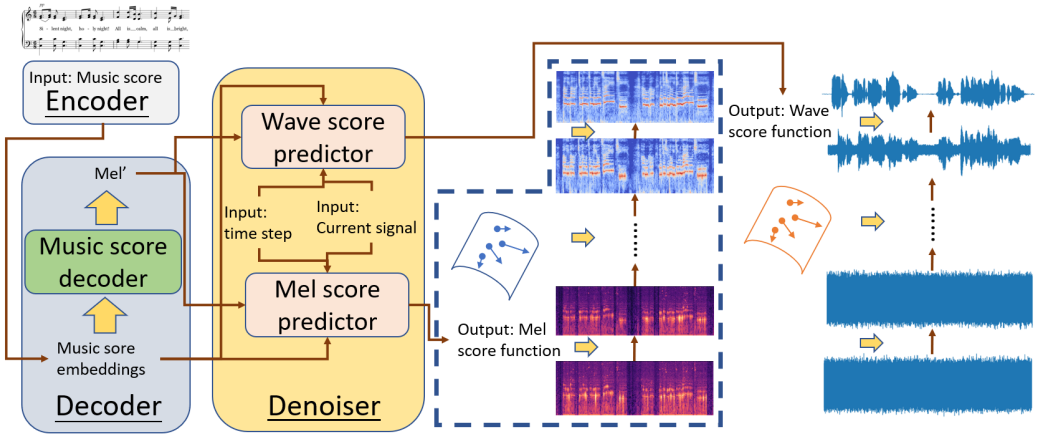

Figure 3: The inference pipeline of RealSinger.

$\mathfrak{S}_{\theta_*}(\mathbf{x}(t), t, M', MS)$. Figure 3 shows the how RealSinger sings the music score. The encoder first converts the music score into embeddings MS, and the decoder will transcribe it into the approximated mel-spectrogram M'. Then with the denoiser, the Langevin dynamics (Welling & Teh, 2011) or the reverse-time Itô SDE (9) can be used to generate the mel-spectrogram corresponding to certain music score. The reverse-time SDE (9) can be solved and used to generate target mel-spectrogram data.

In this paper, we use a strategy that combines reverse-time diffusion sampling and Langevin dynamics (Welling & Teh, 2011), that means at each time step, the reverse-time SDE

$$\mathbf{x}(k\Delta t) = \mathbf{x}(k\Delta t + \Delta t) + \frac{1}{2}\beta(k\Delta t + \Delta t)\left[\mathbf{x}(k\Delta t + \Delta t) - M\right] + \beta(k\Delta t + \Delta t)\mathfrak{S}_{\theta_*}(\mathbf{x}(k\Delta t + \Delta t), k\Delta t + \Delta t, M', S)\Delta t + \sqrt{\beta(k\Delta t + \Delta t)}\xi(k\Delta t)$$

is used to predict the mel-spectrogram corresponding to the music score first, then the Langevin dynamics

$$\mathbf{x}(k\Delta t) \leftarrow \mathbf{x}(k\Delta t) + \epsilon_k \mathfrak{S}_{\theta_*}(\mathbf{x}(k\Delta t), k\Delta t, M', MS) + \sqrt{2\epsilon_k}\xi(k\Delta t) \tag{10}$$

is used to modify the first predicted result.

3.4 End-to-end SVS

As suggested in Figure 1, general SVS is a connected system with two components. RealSinger is flexible enough to support joint optimization of these two components as shown in Figure 2. This

joint optimization is also called end-to-end (E2E) SVS, in which mel-spectrogram score prediction is not needed, and thus can avoid the accumulation error caused by non-perfectly generated mel-spectrogram. In E2E RealSinger, we employ the variance exploding SDE (Song et al., 2021),

$$\begin{cases} d\mathbf{X} = \sigma_0(\frac{\sigma_1}{\sigma_0})^t\sqrt{2\log\frac{\sigma_1}{\sigma_0}}d\mathbf{W}, \quad \sigma_0 = 0.01 < \sigma_1 = 50, \\ \mathbf{X}(0) = \mathbf{x}(0) \sim \int p_{wave}(\mathbf{x})\mathcal{N}(\mathbf{x}(0); \mathbf{x}, \sigma_0^2\mathbf{I})d\mathbf{x}, \end{cases} \tag{11}$$

which has been proved capable to generate human-level waveform. The objective and training algorithm of E2E RealSinger are almost the same as in Algorithm 1.

3.5 IMPLEMENTATIONS

The music score contains lyrics, notes and duration, where the lyrics are first converted into phoneme (and further to integer) sequences, and the notes and duration are in sequences of positive float. The encoder first encodes the phoneme sequences containing the sinusoid position encoding information, and then sends the encoding to the 4-layer feed-forward Transformer (FFT) block to obtain the feature map of the encoded phone. Then the notes and duration will be encoded into embeddings by using the standard Pytorch's (Paszke et al., 2019) 'nn.embedding' function, and added to the feature map of lyrics. The duration is used to extend the feature map to the length of the actual mel-spectrogram of the singing voice. The feature map of the music score will be sent to the decoder and denoiser as the conditional input. The decoder is to convert score embeddings into the singing mel-spectrogram. It is also composed of 4 layers of FFT.

There are two additional inputs for the denoiser, one is the conditional time step information, the other is the mel-spectrogram of singing voice or Gaussian noise (clean mel-spectrogram during training, noise during inference). The singing voice's mel-spectrogram will be encoded through several linear layers and SILU (SIgmoid Linear Unit module) layers (Ramachandran et al., 2017); the time step information will be coded through a module called Gaussian Fourier projection (GFP) (Ren et al., 2020a), and then it will be added to the mel-spectrogram representations. Then the feature maps will be sent to the key module of denoiser, several dilated residual blocks, each of which consists of two 1D convolution layers and 'CHUNK's. It has two outputs, one is status information, which is sent to the next residual block, and the other is part of the score information as output. While the output of the encoder, which is musical score embeddings, will be sent to each residual block as the main ingredient controlling the output of the denoiser. Finally, the outputs of all residual blocks are averaged, and after the two convolution layers are output, the score of the singing mel-spectrogram in the distribution is obtained.

In end-to-end RealSinger, for the structure of wave score prediction network, the input is the wave to be generated or the noise, and the conditional input has the output singing mel-spectrogram from decoder, the music score embedding from the encoder, and time t. The output is the score at time t. All the input require preprocessing processes. The preprocessing of the wave is through a convolution layer; the preprocessing of mel-spectrogram and music score embedding is based on the upsampling by two transposed convolution layers. After all inputs are preprocessed, they will be sent to the most critical module, which is of several serially connected dilated residual blocks. The main input of the dilated residual block is the wave, and the step time condition, music score embeddings and mel-spectrogram condition will be input into these dilated residual blocks one after another, and added to the feature map after the transformation of the wave signal. Similarly, there are two outputs of each dilated residual block, one is the state, which is used for input to the next residual block, and the other is the final output. The advantage of this is the ability to synthesize information of different granularities. Finally, the outputs of all residual blocks are summed and then pass through two convolution layers as the final output score.

4 RELATED WORK

RealSinger falls in the intersection of SVS and diffusion probability models. The earliest SVS system uses pre-recorded short-waveform singing units selected from the database to stitch together into a complete target singing voice (Macon et al., 1997; Kenmochi & Ohshita, 2007). This method has two defects, one is that it requires a large-scale database to provide enough complete singing units, and the other is that this method can not ensure that all singing units be connected smoothly.

To address these two deficiencies, statistical parametric methods, such as Hidden Markov Models (HMM), are proposed for singing voice generation (Saino et al., 2006; Oura et al., 2010). However, the synthesis system based on HMM have the over-smoothing effect, and the quality of the synthesized sound cannot reach the naturalness of the real singing voice.

In recent years, SVS has taken a big step forward in naturalness thanks to the powerful fitting and representation capabilities of deep learning. For example, various network structures, such as feed-forward neural network, long short-term memory (LSTM), convolutional neural network (CNN), and recurrent neural network (RNN), have been successively applied to SVS and shown to outperform connected or HMM-based systems. Especially in the past three years, many successful systems have appeared in the network structure and the construction of the singing database. For example, DeepSinger (Ren et al., 2020b) builds singing data by mining data from music websites, and builds an SVS system from scratch based on this data. SingGAN (Chen et al., 2021) introduces source excitation and AFL filter in the generator, which effectively alleviates the glitches in singing vocal synthesis. At the same time, it introduces a multi-band discriminator with additional frequency domain loss and sub-band feature matching loss to achieve stable training and high-frequency reconstruction. Multi-Singer (Huang et al., 2021) uses GAN to model the singing voice of unknown singers. U-Singer (Kim et al., 2022) expresses emotional intensity by controlling subtle changes in pitch, energy, and phoneme duration. Singing-Tacotron (Wang et al., 2022b) is an end-to-end SVS model with a global duration-controlled attention mechanism, aiming to make the attention mechanism controlled by duration information and improve the naturalness of synthesis. Deep performer (Dong et al., 2022) proposes polyphonic mixers to align encoders and decoders with polyphonic inputs, and also proposes note-by-note positional encoding to provide fine-grained tuning for synthesis models. WeSinger (Zhang et al., 2022) proposes, a multi-scale rhythm loss and a progressive pitch-weighted decoder loss, and a data augmentation method for variable duration segmentation to bridge the gap between accuracy and naturalness.

Very recently, diffusion probability modelling has shown much better performance on image and audio generation compared with GAN, flow or VAE. The earliest source of the ideas for diffusion-based generative modelling should be the pioneering change of data estimation problem into the estimation of the gradient of log of the data distribution density by Hyvärinen & Dayan (2005), thus greatly simplifying the original problem. Another source is the pioneering use of a diffusion Markov chain by Sohl-Dickstein et al. (2015) to diffuse the structure of the image data into a simple distribution, and another opposite diffusion Markov chain to generate images in the target distribution from the simple distribution. These two primitive ideas have been carried forward. Ho et al. (2020) has recently generated very high-quality large-scale natural images with a diffusion Markov chain. DiffWave (Kong et al., 2020) uses a diffusion Markov chain for the vocoder. Song & Ermon (2019) draws on the idea of (Hyvärinen & Dayan, 2005) to estimate the log gradient of the target data distribution density function through a neural network, and then uses the Langevin dynamics to generate large-scale image data in target distribution. Wavegrad (Chen et al., 2020) transplanted the algorithm of Song & Ermon (2019) to vocoders, but in fact the final algorithm is exactly the same as Diffwave (Kong et al., 2020). Immediately afterward, Song et al. (2020) further extended the Markov chain to the continuous case, it became a stochastic differential equation. Ingeniously, the equation unified the two methods of Ho et al. (2020) and Song & Ermon (2019) under one framework, and both became its special cases.

Concurrent works: The most relevant work is DiffSinger (Liu et al., 2021), which proposes a hybrid system using a feed-forward Transformer (FFT) blocks (Ren et al., 2020a) with a shallow diffusion mechanism for fast sampling. Hybrid in DiffSinger is a compromise, as the authors found that the vocals synthesized by the naive denoising diffusion probabilistic models (DDPM) (Ho et al., 2020) had noticeable noise during ventilation. They argue that diffusion deviates from the expected path due to over thousands of steps accumulating random errors. Our RealSinger is a pure continuous diffusion probability model, which controls the diffusion path by adding acoustic feature information as a hint in the SDE. Further, RealSinger supports end-to-end SVS.

5 EXPERIMENTS

5.1 DATASET

The data set we use is PopCS (Liu et al., 2021), a singing voice database from a single qualified female vocalist, with a total of 5.95 hours accompanying phoneme-level aligned music scores. PopCS has 127 Chinese pop songs, which are split into 5,498 song pieces accompanying phoneme-level aligned music scores, randomly divided into 4948/275/275 for training/verification/testing, as the same setting of (Liu et al., 2021). The sampling rate is 22050 Hz.

5.2 EXPERIMENTAL SETUP

In the experiment, for mel-spectrogram, the window length is 512, hop length is 128, and the number of mel channels is 80. We use the Adam (Kingma & Ba, 2014) training algorithm for RealSinger. We have done objective and quantitative evaluations based on mean opinion score (MOS) with other state-of-the-art methods. For RealSinger, we compared with FFT-Singer and DiffSinger (Liu et al., 2021). All experiments were performed on a GeForce RTX 3090 GPU with 24G memory.

For mel-spectrogram in this experiment, the window length is 512, the hop length is 128, and the number of mel channels is 80. Adam (Kingma & Ba, 2014) is used to train RealSinger. Objective and quantitative evaluations have been done with other state-of-the-art methods. RealSinger is compared with FFT-Singer and DiffSinger (Liu et al., 2021). All experiments were performed on a GeForce RTX 3090 GPU with 24G of memory.

In order to compare the singing quality of RealSinger with FFT-Singer and DiffSinger (Liu et al., 2021), a pre-trained HiFi-GAN model (Su et al., 2020) as a vocoder to transform the singing mel-spectrogram into the wave. For FFT-Singer and DiffSinger, we use the same hype-parameters as in Liu et al. (2021).

RealSinger uses 4 FFT layers in the music score encoder and decoder. Both mel and wave score predictors use 20 residual layers. The approximated mel-spectrogram output of the decoder and the music score embeddings are used as the condition input to the score estimation network.

The parameters of generalized VP linear SDE (3) in the experiment are $\beta_0 = 0.05$, $\beta_1 = 20$. For VE linear SDE (11), we have $\sigma_0 = 0.01$, $\sigma_1 = 50$, and the number of time steps $N = 1000$.

5.3 SUBJECTIVE EVALUATION

To verify the naturalness and fidelity of the synthesized singing voice, we randomly selected 100 out of 275 test song pieces for each subject, and then requested the subject to give the synthesized singing voice a MOS score of 0-5. 0 means 'awful' and 5 means 'excellent'. Table 1 shows the model size and MOS of all systems. It can be seen that RealSinger's model is about the same size as DiffSinger, but has a 0.16 MOS improvement. The end-to-end approach will further improve RealSinger's MOS to 4.48, which is very close to the human's singing voice.

Table 1: MOS with 95% confidence in a comparative study of different state-of-the-art SVS methods on the PopCS test set. All methods use a pre-trained HiFi-GAN as vocoder.

Methods	Model size	MOS
Ground truth	-	4.56± 0.07
FFT-Singer	24.3M	3.87± 0.170
DiffSinger	39.3M	4.23± 0.165
RealSinger	39.4M	4.39± 0.160
RealSinger E2E	45.6M	4.48± 0.145

5.4 OBJECTIVE EVALUATION

Mel cepstral distortion (MCD) (Kubichek, 1993) is an objective metric for voice quality assessment. The implementation of MCD from Samuel Broughton[1] is used in this paper. The generated and

ground truth waveforms are converted into mel-cepstral coefficients (MCEPS), and then the MCD between these MCEPS are computed. The smaller of MCD, the smaller the distortion of the sound, that is, the higher quality of the singing voice. Table 2 shows the MCD of all systems. It can be seen that RealSinger can produce singing voices with minimal disturbance compared to real human voices.

Table 2: MCD in a comparative study of different state-of-the-art SVS methods on PopCS test set.

Methods	MCD (dB)
Ground truth mel + HiFi-GAN	3.81
FFT-Singer + HiFiGAN	6.08
DiffSinger + HiFiGAN	5.65
RealSinger + HiFiGAN	5.39
RealSinger E2E	5.12

5.5 Ablation study

We conducted ablation studies to prove the effectiveness of RealSinger, including

- Naive RealSinger: Use the original VP SDE (Song et al., 2021), that means without the constant M in the SDE (3).
- Shallow RealSinger: Use the shallow mechanism similar to (Liu et al., 2021).
- Super RealSinger: Use 10,000 steps in diffusion.

The results are shown in Table 3. Removal of the constant M in the SDE (3) results in 0.38 MOS lower than the baseline (RealSinger), which indicates that the benchmark at the start of diffusion affects the synthesis quality. It can also be seen that the more steps of diffusion, the better the performance, especially Super RealSinger (4.54 MOS) basically reaches the level of human singing voices (4.56 MOS).

Table 3: MOS with 95% confidence in the ablation study.

Methods	MOS
Ground truth	4.56 ± 0.07
Naive RealSinger	4.01 ± 0.185
Shallow RealSinger	4.19 ± 0.172
Super RealSinger	4.54 ± 0.14
RealSinger	4.39 ± 0.160
RealSinger E2E	4.48 ± 0.143

6 Conclusion

In this paper, we propose RealSinger, a novel SVS system based on linear SDEs. Given a musical score, RealSinger can continuously transform Gaussian noise into corresponding singing mel-spectrogram and wave through reverse-time linear SDE and Langevin dynamic. RealSinger is a pure diffusion-based method, it use neural networks to predict the mel and wave score, which is the gradient of the log probability density at a specific time. For RealSinger, we designed the corresponding effective score prediction networks. Subjective and objective evaluation show that the RealSinger can achieve the state-of-the-art.

References

Brian DO Anderson. Reverse-time diffusion equation models. *Stochastic Processes and their Applications*, 12(3):313–326, 1982.

[1]https://github.com/SamuelBroughton/Mel-Cepstral-Distortion

Feiyang Chen, Rongjie Huang, Chenye Cui, Yi Ren, Jinglin Liu, and Zhou Zhao. Singgan: Generative adversarial network for high-fidelity singing voice generation. *arXiv preprint arXiv:2110.07468*, 2021.

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.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 34, 2021.

Hao-Wen Dong, Cong Zhou, Taylor Berg-Kirkpatrick, and Julian McAuley. Deep performer: Score-to-audio music performance synthesis. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 951–955. IEEE, 2022.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems 27*, volume 27, pp. 2672–2680, 2014.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *arXiv preprint arXiv:2006.11239*, 2020.

Rongjie Huang, Feiyang Chen, Yi Ren, Jinglin Liu, Chenye Cui, and Zhou Zhao. Multi-singer: Fast multi-singer singing voice vocoder with a large-scale corpus. In *Proceedings of the 29th ACM International Conference on Multimedia*, pp. 3945–3954, 2021.

Aapo Hyvärinen and Peter Dayan. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(4), 2005.

Hideki Kenmochi and Hayato Ohshita. Vocaloid-commercial singing synthesizer based on sample concatenation. In *Interspeech*, volume 2007, pp. 4009–4010, 2007.

Sungjae Kim, Kihyun Na, Choonghyeon Lee, Jehyeon An, and Injung Kim. U-singer: Multi-singer singing voice synthesizer that controls emotional intensity. *arXiv preprint arXiv:2203.00931*, 2022.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

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.

Robert Kubichek. Mel-cepstral distance measure for objective speech quality assessment. In *Proceedings of IEEE pacific rim conference on communications computers and signal processing*, volume 1, pp. 125–128. IEEE, 1993.

Jinglin Liu, Chengxi Li, Yi Ren, Feiyang Chen, Peng Liu, and Zhou Zhao. Diffsinger: Singing voice synthesis via shallow diffusion mechanism. *arXiv preprint arXiv:2105.02446*, 2021.

Peng Liu, Yuewen Cao, Songxiang Liu, Na Hu, Guangzhi Li, Chao Weng, and Dan Su. Vara-tts: Non-autoregressive text-to-speech synthesis based on very deep vae with residual attention. *arXiv preprint arXiv:2102.06431*, 2021.

Michael W Macon, Leslie Jensen-Link, James Oliverio, Mark A Clements, and E Bryan George. A singing voice synthesis system based on sinusoidal modeling. In *1997 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pp. 435–438. IEEE, 1997.

Keiichiro Oura, Ayami Mase, Tomohiko Yamada, Satoru Muto, Yoshihiko Nankaku, and Keiichi Tokuda. Recent development of the hmm-based singing voice synthesis system—sinsy. In *Seventh ISCA Workshop on Speech Synthesis*, 2010.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, volume 32, pp. 8026–8037, 2019.

Vadim Popov, Ivan Vovk, Vladimir Gogoryan, Tasnima Sadekova, and Mikhail Kudinov. Grad-tts: A diffusion probabilistic model for text-to-speech. *arXiv preprint arXiv:2105.06337*, 2021.

Prajit Ramachandran, Barret Zoph, and Quoc V. Le. Swish: a self-gated activation function. *arXiv: Neural and Evolutionary Computing*, 2017.

Yi Ren, Chenxu Hu, Xu Tan, Tao Qin, Sheng Zhao, Zhou Zhao, and Tie-Yan Liu. Fastspeech 2: Fast and high-quality end-to-end text to speech. *arXiv preprint arXiv:2006.04558*, 2020a.

Yi Ren, Xu Tan, Tao Qin, Jian Luan, Zhou Zhao, and Tie-Yan Liu. Deepsinger: Singing voice synthesis with data mined from the web. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1979–1989, 2020b.

Keijiro Saino, Heiga Zen, Yoshihiko Nankaku, Akinobu Lee, and Keiichi Tokuda. An hmm-based singing voice synthesis system. In *Ninth International Conference on Spoken Language Processing*, 2006.

Simo Särkkä and Arno Solin. *Applied stochastic differential equations*, volume 10. Cambridge University Press, 2019.

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

Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *arXiv preprint arXiv:1907.05600*, 2019.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*, 2021.

Jiaqi Su, Zeyu Jin, and Adam Finkelstein. Hifi-gan: High-fidelity denoising and dereverberation based on speech deep features in adversarial networks. *arXiv preprint arXiv:2006.05694*, 2020.

Rafael Valle, Kevin Shih, Ryan Prenger, and Bryan Catanzaro. Flowtron: an autoregressive flow-based generative network for text-to-speech synthesis. *arXiv preprint arXiv:2005.05957*, 2020.

Pascal Vincent. A connection between score matching and denoising autoencoders. *Neural computation*, 23(7):1661–1674, 2011.

Shoutong Wang, Jinglin Liu, Yi Ren, Zhen Wang, Changliang Xu, and Zhou Zhao. Mr-svs: Singing voice synthesis with multi-reference encoder. *arXiv preprint arXiv:2201.03864*, 2022a.

Tao Wang, Ruibo Fu, Jiangyan Yi, Jianhua Tao, and Zhengqi Wen. Singing-tacotron: Global duration control attention and dynamic filter for end-to-end singing voice synthesis. *arXiv preprint arXiv:2202.07907*, 2022b.

Yuxuan Wang, RJ Skerry-Ryan, Daisy Stanton, Yonghui Wu, Ron J Weiss, Navdeep Jaitly, Zongheng Yang, Ying Xiao, Zhifeng Chen, Samy Bengio, et al. Tacotron: Towards end-to-end speech synthesis. *arXiv preprint arXiv:1703.10135*, 2017.

Max Welling and Yee W. Teh. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th International Conference on Machine Learning*, pp. 681–688, 2011.

Zewang Zhang, Yibin Zheng, Xinhui Li, and Li Lu. Wesinger: Data-augmented singing voice synthesis with auxiliary losses. *arXiv preprint arXiv:2203.10750*, 2022.