Learning the Beauty in Songs: Neural Singing Voice Beautifier

Jinglin Liu and **Chengxi Li** and **Yi Ren** and **Zhiying Zhu** and **Zhou Zhao**[*]
{jinglinliu,chengxili,rayeren,zhyingzh,zhaozhou}@zju.edu.cn
Zhejiang University

Abstract

We are interested in a novel task, singing voice beautification (SVB). Given the singing voice of an amateur singer, SVB aims to improve the intonation and vocal tone of the voice, while keeping the content and vocal timbre. Current automatic pitch correction techniques are immature, and most of them are restricted to intonation but ignore the overall aesthetic quality. Hence, we introduce Neural Singing Voice Beautifier (NSVB), the first generative model to solve the SVB task, which adopts a conditional variational autoencoder as the backbone and learns the latent representations of vocal tone. In NSVB, we propose a novel time-warping approach for pitch correction: Shape-Aware Dynamic Time Warping (SADTW), which ameliorates the robustness of existing time-warping approaches, to synchronize the amateur recording with the template pitch curve. Furthermore, we propose a latent-mapping algorithm in the latent space to convert the amateur vocal tone to the professional one. To achieve this, we also propose a new dataset containing parallel singing recordings of both amateur and professional versions. Extensive experiments on both Chinese and English songs demonstrate the effectiveness of our methods in terms of both objective and subjective metrics. Audio samples are available at `https://neuralsvb.github.io`. Codes: `https://github.com/MoonInTheRiver/NeuralSVB`.

1 Introduction

The major successes of the artificial intelligent singing voice research are primarily in Singing Voice Synthesis (SVS) (Lee et al., 2019; Blaauw and Bonada, 2020; Ren et al., 2020; Lu et al., 2020; Liu et al., 2021a) and Singing Voice Conversion (SVC) (Sisman and Li, 2020; Li et al., 2021; Wang et al., 2021a). However, the Singing Voice Beautification (SVB) remains an important and challenging endeavor for researchers. SVB aims to improve the intonation[1] and the vocal tone of the voice, while keeping the content and vocal timbre[2]. SVB is extensively required both in the professional recording studios and the entertainment industries in our daily life, since it is impractical to record flawless singing audio.

Nowadays in real-life scenarios, SVB is usually performed by professional sound engineers with adequate domain knowledge, who manipulate commercial vocal correction tools such as Melodyne[3] and Autotune[4] (Yong and Nam, 2018). Most current automatic pitch correction works are shown to be an attractive alternative, but they may 1) show weak alignment accuracy (Luo et al., 2018) or pitch accuracy (Wager et al., 2020); 2) cause the tuned recording and the reference recording to be homogeneous in singing style (Yong and Nam, 2018). Besides, they typically focus on the intonation but ignore the overall aesthetic quality (audio quality and vocal tone) (Rosenzweig et al., 2021; Zhuang et al., 2021).

To tackle these challenges, we introduce Neural Singing Voice Beautifier (NSVB), the first generative model to solve the SVB task, which adopts a Conditional Variational AutoEncoder (CVAE) (Kingma and Welling, 2014; Sohn et al., 2015) as the backbone to generate high-quality audio and learns the latent representation of vocal tone. In NSVB, we dichotomize the SVB task into pitch correction and vocal tone improvement: 1) To correct the intonation, a straightforward method is aligning the amateur recording with the template pitch curve, and then putting them together to resynthesize a new singing sample. Previous

[*] Corresponding Author

[1] Intonation refers to the accuracy of pitch in singing.

[2] The differences between the vocal tone and vocal timbre is that: the former represents one's skills of singing, such as airflow controlling ability, muscle strength of vocal folds and vocal placement; the latter represents the identical, overall sound of one's vocal.

[3] `https://www.celemony.com/en/start`

[4] `https://www.antarestech.com/`

Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 7970 - 7983
May 22-27, 2022

works (Wada et al., 2017; Luo et al., 2018) implemented this by figuring out the alignment through Dynamic Time Warping (DTW) (Müller, 2007) or Canonical Time Warping (CTW) (Zhou and Torre, 2009). We propose a novel Shape-Aware DTW algorithm, which ameliorates the robustness of existing time-warping approaches by considering the shape of the pitch curve rather than low-level features when calculating the optimal alignment path. 2) To improve the vocal tone, we propose a latent-mapping algorithm in the latent space, which converts the latent variables of the amateur vocal tone to those of the professional ones. This process is optimized by maximizing the log-likelihood of the converted latent variables. To retain the vocal timbre during the vocal tone mapping, we also propose a new dataset named PopBuTFy containing parallel singing recordings of both amateur and professional versions. Besides, thanks to the autoencoder structure, NSVB inherently supports semi-supervised learning, where the additional unpaired, unlabeled[5] singing data could be leveraged to facilitate the learning of the latent representations. Extensive experiments on both Chinese and English songs show that NSVB outperforms previous methods by a notable margin, and each component in NSVB is effective, in terms of both objective and subjective metrics. The main contributions of this work are summarized as follows:

- We propose the first generative model NSVB to solve the SVB task. NSVB not only corrects the pitch of amateur recordings, but also generates the audio with high audio quality and improved vocal tone, to which previous works typically pay little attention.

- We propose Shape-Aware Dynamic Time Warping (SADTW) algorithm to synchronize the amateur recording with the template pitch curve, which ameliorates the robustness of the previous time-warping algorithm.

- We propose a latent-mapping algorithm to convert the latent variable of the amateur vocal tone to the professional one's, and contribute a new dataset PopBuTFyto train the latent-mapping function.

- We design NSVB as a CVAE model, which supports the semi-supervised learning to leverage unpaired, unlabeled singing data for better performance.

2 Related Works

2.1 Singing Voice Conversion

Singing Voice Conversion (SVC) is a sub-task of Voice Conversion (VC) (Berg-Kirkpatrick and Klein, 2015; Serrà et al., 2019; Popov et al., 2021; Liu et al., 2021b), which transforms the vocal timbre (or singer identity) of one singer to that of another singer, while preserving the linguistic content and pitch/melody information (Li et al., 2021). Mainstream SVC models can be grouped into three categories (Zhao et al., 2020): 1) parallel spectral feature mapping models, which learn the conversion function between source and target singers relying on parallel singing data (Villavicencio and Bonada, 2010; Kobayashi et al., 2015; Sisman et al., 2019); 2) Cycle-consistent Generative Adversarial Networks (CycleGAN) (Zhu et al., 2017; Kaneko et al., 2019), where an adversarial loss and a cycle-consistency loss are concurrently used to learn the forward and inverse mappings simultaneously (Sisman and Li, 2020); 3) encoder-decoder models, such as PPG-SVC (Li et al., 2021), which leverage a singing voice synthesis (SVS) system for SVC (Zhang et al., 2020), and auto-encoder (Qian et al., 2019a; Wang et al., 2021b; Yuan et al., 2020) based SVC (Wang et al., 2021a). The models of the latter two categories can be utilized with non-parallel data. In our work, we aim to convert the intonation and the vocal tone while keeping the content and the vocal timbre, which is quite different from the SVC task.

2.2 Automatic Pitch Correction

Automatic Pitch Correction (APC) works attempt to minimize the manual effort in modifying the flawed singing voice (Yong and Nam, 2018). Luo et al. (2018) propose Canonical Time Warping (CTW) (Zhou and Torre, 2009; Zhou and De la Torre, 2012) which aligns amateur singing recordings to professional ones according to the pitch curves only. Wager et al. (2020) propose a data-driven approach to predict pitch shifts depending on both amateur recording and its accompaniment. Rosenzweig et al. (2021) propose a pitch shift method for Cappella recordings. Zhuang et al. (2021) propose a pitch-controllable SVS system to resynthesize the audio with correctly predicted pitch curves. Besides modifying pitch, Yong and

[5] "unpaired, unlabeled" means the recordings sung by any people, in any vocal tone without label.

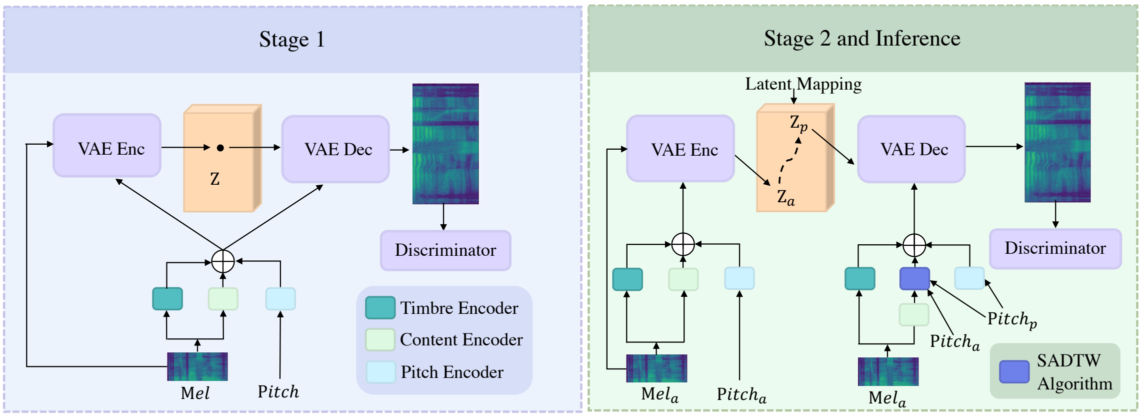


Figure 1: The overview of NVSB. The training process consists of 2 stages, and the second stage shares the same pipeline with the inference stage. “VAE Enc” means the encoder of CVAE; “VAE Dec” means the decoder of CVAE; “Mel” means the mel-spectrogram; “$\mathbf{z}$” means the latent variable of the vocal tone; the “a”/“p” subscript means the amateur/professional version.

Nam (2018) propose to modify pitch and energy information to improve the singing expressions of an amateur singing recording. However, this method heavily relies on a reference recording, causing the tuned recording and the reference recording to be homogeneous in singing style (Zhuang et al., 2021). Our work adopts the non-parametric and data-free pitch correction method like Luo et al. (2018), but improves the accuracy of alignment.

3 Methdology

In this section, we describe the overview of NSVB, which is shown in Figure 1. At Stage 1 in the figure, we reconstruct the input mel-spectrogram through the CVAE backbone (Section 3.1) based on the pitch, content and vocal timbre conditions extracted from the input by the pitch encoder, content encoder and timbre encoder, and optimize the CVAE by maximizing evidence lower bound and adversarial learning. At Stage 2/Inference in the figure, firstly we infer the latent variable $\mathbf{z}_a$ based on the amateur conditions; secondly we prepare the amateur content vectors aligned with the professional pitch by SADTW algorithm (Section 3.2); thirdly we map $\mathbf{z}_a$ to $\mathbf{z}_p$ by the latent-mapping algorithm (Section 3.3); finally, we mix the professional pitch, the aligned amateur content vectors, and the amateur vocal timbre to obtain a new condition, which is leveraged along with the mapped $\mathbf{z}_p$ by the decoder of CVAE to generate a new beautified mel-spectrogram. The training/inference details and model structure of each component in NSVB are described in Section 3.4 and Section 3.5.

3.1 Conditional Variational Generator with Adversarial Learning

As shown in Figure 2, to generate audio with high quality and learn the latent representations of vocal tone, we introduce a Conditional Variational AutoEncoder (CVAE) (Kingma and Welling, 2014; Sohn et al., 2015) as the mel-spectrogram generator, with the optimizing objective of maximizing the evidence lower bound (ELBO) of the intractable marginal log-likelihood of mel-spectrogram $\log p_\theta(\mathbf{x}|c)$:

$$\log p_\theta(\mathbf{x}|c) \geq \mathbf{ELBO}(\phi,\theta) \equiv$$
$$E_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x},c)}\left[\log p_\theta(\mathbf{x}|\mathbf{z},c) - \log\frac{q_\phi(\mathbf{z}|\mathbf{x},c)}{p(\mathbf{z})}\right],$$

where $\mathbf{x}$, c, $\mathbf{z}$ denote the input/output mel-spectrogram, the mix of content, vocal timbre and pitch conditions, and the latent variable representing the vocal tone respectively; ϕ and θ denote the model parameters of CVAE encoder and CVAE decoder; $q_\phi(\mathbf{z}|\mathbf{x},c)$ is the posterior distribution approximated by the CVAE encoder; $p_\theta(\mathbf{x}|\mathbf{z},c)$ is the likelihood function that generates mel-spectrograms given latent variable $\mathbf{z}$ and condition c; $p(\mathbf{z})$ is the prior distribution of the latent variables $\mathbf{z}$, and we choose the standard normal distribution as $p(\mathbf{z})$ for simplification. Furthermore, to address the over-smoothing problem (Qian et al., 2019b) in CVAE, we utilize an adversarial discriminator ($\mathcal{D}$) (Mao et al., 2017) to refine the output mel-spectrogram:

$$L_{adv}(\phi,\theta) = \mathbb{E}[(\mathcal{D}(\widetilde{\mathbf{x}})-1)^2],$$
$$L_{adv}(\mathcal{D}) = \mathbb{E}[(\mathcal{D}(\mathbf{x})-1)^2] + \mathbb{E}[\mathcal{D}(\widetilde{\mathbf{x}})^2], \quad (1)$$

where $\mathbf{x}$ is the ground-truth and $\widetilde{\mathbf{x}}$ is the output of CVAE. The descriptions for the model structure of each component are in Section 3.5.

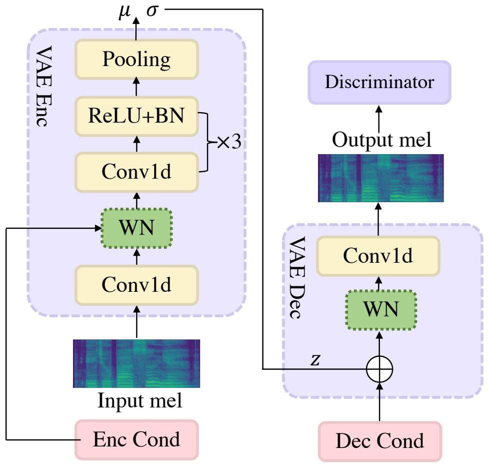


Figure 2: The CVAE backbone in NSVB. "Enc/Dec Cond" means the conditions for the encoder/decoder; "Conv1d" means the 1-D convolutional layer; "Pooling" means the average pooling layer; μ and σ represent the approximated mean and log scale standard deviation parameters in the posterior Gaussian distribution; $\mathbf{z}$ is the sampled latent variable.

3.2 Shape-Aware Dynamic Time Warping

To implement the pitch correction, a straightforward method is aligning the amateur recording with the template pitch curve, and then concatenating them to resynthesize a new singing sample with improved intonation. Since the source pitch curve of amateur recordings and template one show a high degree of natural correlation along the time axis, applying a proper time-warping algorithm on them is crucial. However, original DTW (Müller, 2007) could result in a poor alignment when certain parts of the axis move to higher frequencies, and other parts to lower ones, or vice versa (Sundermann and Ney, 2003). Luo et al. (2018) adopt an advanced algorithm CTW (Zhou and Torre, 2009), which combines the canonical correlation analysis (CCA) and DTW to extract the feature sequences of two pitch curves, and then apply DTW on them. However, the alignment accuracy of CTW leaves much to be desired.

We elaborate a non-parametric and data-free algorithm, Shape-Aware DTW (SADTW), based on the prior knowledge that the source pitch curve and the template one have analogous local shape contours. Specifically, we replace the Euclidean distance in the original DTW distance matrix with the shape context descriptor distance. The shape context descriptor of a time point f_i in one pitch curve is illustrated in Figure 3. Inspired by (Mori et al., 2005), we divide the data points around f_i into $m * n$ bins by m time windows and n angles. We calculate the number of all points falling in the k-th bin. Then the descriptor for f_i is defined as the histogram $h_i \in \mathcal{R}^{m*n}$:

$$h_i(k) = |\{f_j \neq f_i, f_j \in bin(k)\}|,$$

where $|\cdot|$ means the cardinality of a set. This histogram represents the distribution over relative positions, which is a robust, compact and discriminative descriptor. Then, it is natural to use the $\mathcal{X}^2$-test statistic on this distribution descriptor as the "distance" of two points f_a and f_p:

$$C(a,p) = \frac{1}{2}\sum_{k=1}^{m*n}\frac{[h_a(k) - h_p(k)]^2}{h_a(k) + h_p(k)},$$

where h_a and h_p are the normalized histograms corresponding to the point f_a from the amateur pitch curve and the point f_p from the template pitch curve. $C(a,p)$ ranges from 0 to 1. Finally, we run DTW on the distance matrix C to obtain the alignment with least distance cost between two curves.

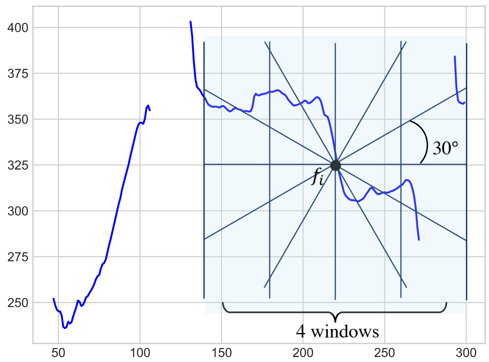


Figure 3: The shape descriptor in SADTW. The blue curve represents pitch; the horizontal axis means time; the vertical axis means F0-frequency. There are $m = 4$ windows, $n = 6$ angles to divide neighbor points of f_i.

3.3 Latent-mapping Algorithm

Define a pair of mel-spectrograms $(\mathbf{x}_a, \mathbf{x}_p)$: the contents of $\mathbf{x}_a$ and $\mathbf{y}_p$ are the same sentence of a song from the same singer[6], who sings these two recordings using the amateur tone and the professional tone respectively. Given the CVAE model,

[6] The singers all major in vocal music.

we can infer the posterior distribution $q_\phi(\mathbf{z}_a|\mathbf{x}_a, c_a)$ and $q_\phi(\mathbf{z}_p|\mathbf{x}_p, c_p)$ corresponding to $\mathbf{x}_a$ and $\mathbf{x}_p$ through the encoder of CVAE. To achieve the conversion of vocal tone, we introduce a mapping function $\mathcal{M}$ to convert the latent variables from $q_\phi(\mathbf{z}_a|\mathbf{x}_a, c_a)$ to $q_\phi(\mathbf{z}_p|\mathbf{x}_p, c_p)$. Concretely, we sample a latent variable of amateur vocal tone $\mathbf{z}_a$ from $q_\phi(\mathbf{z}_a|\mathbf{x}_a, c_a)$, and map $\mathbf{z}_a$ to $\mathcal{M}(\mathbf{z}_a)$. Then, $\mathcal{M}$ can be optimized by minimizing the negative log-likelihood of $\mathcal{M}(\mathbf{z}_a)$:

$$L_{map1}(\mathcal{M}) = -\log q_\phi(\mathcal{M}(\mathbf{z}_a)|\mathbf{x}_p, c_p).$$

Define $\hat{c}_p$ as the mix of 1) the content vectors from the amateur recording aligned by SADTW, 2) vocal timbre embedding encoded by timbre encoder, and 3) template pitch[7] embeddings encoded by pitch encoder. To make sure the converted latent variable could work well together with $\hat{c}_p$ to generate a high-quality audio sample (with the correct pitch and improved vocal tone), we send $\mathcal{M}(\mathbf{z}_a)$ to the CVAE decoder to generate $\hat{\mathbf{x}}$, and propose an additional loss:

$$L_{map2}(\mathcal{M}) = \|\hat{\mathbf{x}} - \mathbf{x}_p\|_1 + \lambda(\mathcal{D}(\hat{\mathbf{x}}) - 1)^2,$$

where $\mathcal{D}$ has been optimized by Eq. (1); λ is a hyper-parameter.

3.4 Training and Inference

There are two training stages for NSVB: in the first training stage, we optimize CVAE by minimizing the following loss function

$$L(\phi, \theta) = -\mathbf{ELBO}(\phi, \theta) + \lambda L_{adv}(\phi, \theta),$$

and optimize the discriminator ($\mathcal{D}$) by minimizing Eq. (1). Note that, the first stage is the reconstruction process of mel-spectrograms, where any unpaired, unlabeled singing data beyond PopBuTFy could be leveraged to facilitate the learning of the latent representations. In the second training stage, we optimize $\mathcal{M}$ on the parallel dataset PopBuTFy by minimizing the following loss function

$$L(\mathcal{M}) = L_{map1}(\mathcal{M}) + L_{map2}(\mathcal{M}).$$

ϕ, θ, and $\mathcal{D}$ are not optimized in this stage.

In inference, the encoder of CVAE encodes $\mathbf{x}_a$ with the condition c_a to predict $\mathbf{z}_a$. Secondly, we map $\mathbf{z}_a$ to $\mathcal{M}(\mathbf{z}_a)$, and run SADTW to align the amateur recordings with the template pitch curve. The template pitch curve can be derived from a reference recording with good intonation or a pitch predictor with the input of music notes. Then, we obtain $\hat{c}_p$ defined in Section 3.3 and send $\mathcal{M}(\mathbf{z}_a)$ together with $\hat{c}_p$ in the decoder of CVAE to generate $\hat{\mathbf{x}}$. Finally, by running a pre-trained vocoder conditioned on $\hat{\mathbf{x}}$, a new beautified recording is produced.

3.5 Model Structure

The encoder of CVAE consists of a 1-D convolutional layer (stride=4), an 8-layer WaveNet structure (Oord et al., 2016; Rethage et al., 2018) and 3 1-D convolutional layers (stride=2) with ReLU activation function and batch normalization followed by a mean pooling, which outputs the mean and log scale standard deviation parameters in the posterior distribution of $\mathbf{z}$. The decoder of CVAE consists of a 4-layer WaveNet structure and a 1-D convolutional layer, which outputs the mel-spectrogram with 80 channels. The discriminator adopts the same structure as (Wu and Luan, 2020), which consists of multiple random window discriminators. The latent-mapping function is composed of 2 linear layers to encode the vocal timbre as the mapping condition, and 3 linear layers to map $\mathbf{z}_a$. The pitch encoder is composed of 3 convolutional layers. In addition, given a singing recording, 1) to obtain its content vectors, we train an Automatic Speech Recognition (ASR) model based on Conformer (Gulati et al., 2020) with both speech and singing data, and extract the hidden states from the ASR encoder (viewed as the content encoder) output as the linguistic content information, which are also called phonetic posterior-grams (PPG); 2) to obtain the vocal timbre, we leverage the open-source API resemblyzer[8] as the timbre encoder, which is a deep learning model designed for speaker verification (Wan et al., 2018), to extract the identity information of a singer. More details of model structure can be found in Appendix A.

4 Experiments

4.1 Experimental Setup

In this section, we first introduce PopBuTFy, the dataset for SVB, and then describe the implementation details in our work. Finally, we explain the evaluation method we adopt in this paper.

[7] During training, template pitch is extracted from the waveform corresponding to $\mathbf{x}_p$.

[8] https://github.com/resemble-ai/Resemblyzer

Dataset Since there is no publicly available high-quality, unaccompanied and parallel singing dataset for the SVB task, we collect and annotate a dataset containing both Chinese Mandarin and English pop songs: PopBuTFy. To collect PopBuTFy for SVB, the qualified singers majoring in vocal music are asked to sing a song twice, using the amateur vocal tone for one time and the professional vocal tone for another. Note that some of the amateur recordings are sung off-key by one or more semi-tones for the pitch correction sub-task. The parallel setting could make sure that the personal vocal timbre will keep still during the beautification process. In all, PopBuTFy consists of 99 Chinese pop songs ($\sim$10.4 hours in total) from 12 singers and 443 English pop songs ($\sim$40.4 hours in total) from 22 singers. All the audio files are recorded in a professional recording studio by qualified singers, male and female. Every song is sampled at 22050 Hz with 16-bit quantization. We randomly choose 6 songs in Chinese and 18 songs in English (from unseen speakers) for validation and test. For subjective evaluations, we choose 60 samples in the test set from different singers, half in Chinese and English. All testing samples are included for objective evaluations.

Implementation Details We train the Neural Singing Beautifier on a single 32G Nividia V100 GPU with the batch size of 64 sentences for both 100k steps in Stage 1 and Stage 2 respectively. Besides PopBuTFy, we pre-train the ASR model (used for PPG extraction) leveraging the extra speech datasets: AISHELL-3 (Yao Shi et al., 2020) for Chinese and LibriTTS (Zen et al., 2019) for English. For the semi-supervised learning mentioned in Section 1 and Section 3.4, we leverage an internal Chinese singing dataset ($\sim$30 hours without labeled vocal tone) in the first training stage described in Section 3.4 for Chinese experiments. The output mel-spectrograms of our model are transformed into audio samples using a HiFi-GAN vocoder (Kong et al., 2020) trained with singing data in advance. We set the λ metioned in Section 3.3 to 0.1. We transform the raw waveform with the sampling rate 22050 Hz into mel-spectrograms with the frame size 1024 and the hop size 128. We extract F_0 (fundamental frequency) as pitch information from the raw waveform using Parselmouth[9], following Wu and Luan (2020); Blaauw and Bonada (2020); Ren et al. (2020). To obtain the ground truth pitch alignment between the amateur recordings and the professional ones for evaluating the accuracy of pitch alignment algorithm, we run the Montreal Forced Aligner tool (McAuliffe et al., 2017) on all the singing recordings to obtain their alignments to lyrics. Then the ground-truth pitch alignment can be derived since the lyrics are shared in a pair of data in PopBuTFy.

Performance Evaluation We employ both subjective metrics: Mean Opinion Score (MOS), Comparison Mean Opinion Score (CMOS), and an objective metric: Mean Cepstral Distortion (MCD) to evaluate the audio quality on the test-set. Besides, we use F0 Root Mean Square Error (F0 RMSE) and Pitch Alignment Accuracy (PAA) to estimate the pitch correction performance. For audio, we analyze the MOS and CMOS in two aspects: audio quality (naturalness, pronunciation and sound quality) and vocal tone quality. MOS-Q/CMOS-Q and MOS-V/CMOS-V correspond to the MOS/CMOS of audio quality and vocal tone quality respectively. More details about subjective evaluations are placed in Appendix C.

4.2 Main Results

In this section, we conduct extensive experiments to present our proposed model in regard to 1) the performance of pitch conversion; 2) the audio quality and vocal tone quality.

4.2.1 Pitch Correction

Firstly, we provide the comparison among time-warping algorithms in terms of PAA in Table 1. *Normed DTW* means two pitch curves will be normalized before running *DTW* (Müller, 2007); *CTW* means the Canonical Time Warping (Zhou and Torre, 2009), which is used for pitch correction in Luo et al. (2018). It can be seen that, *SADTW* surpasses existing methods by a large margin. We also visualize an alignment example of *DTW*, *CTW*, and *SADTW* in Figure 4.

Secondly, to check whether the amateur recordings are corrected to the good intonation after being beautified by NSVB, we calculate the F0 RMSE metric of the amateur recordings and the audio generated by NSVB, and list the results in Table 2. We can see that F0 RMSE has been improved significantly, which means NSVB successfully achieve pitch correction.

[9] https://github.com/YannickJadoul/Parselmouth

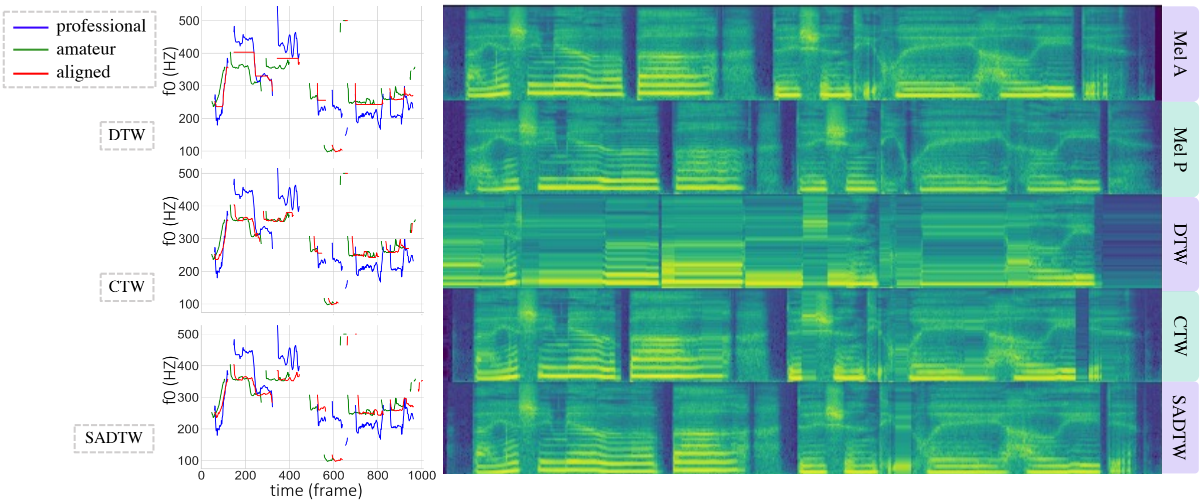


Figure 4: The behavior of DTW, CTW and SADTW. 1) In the left panel of the figure, we align the pitch curve of the amateur recording to the professional one's. It can be seen that DTW perform terribly; CTW fails at many parts; SADTW perform well as expectation. 2) In the right panel of the figure, we use the alignments obtained from these time-warping algorithm on pitch curves to align the amateur mel-spectrogram to the professional one. It shows that only SADTW could provide an alignment which preserves the content information in the amateur recording well and make the aligned result match the professional recording along the time axis.

Table 1: The Pitch Alignment Accuracy of different algorithms on Chinese and English songs.

Algorithm	PAA (%)	
	Chinese	English
DTW	66.94	63.90
Normed DTW	65.19	62.86
CTW	71.35	69.28
SADTW	**79.45**	**78.64**

Table 2: The F0 RMSE of the original amateur audio and the beautified audio on Chinese and English datasets. "GT Amateur" means the ground-truth amateur recordings.

Algorithm	F0 RMSE (Hz)	
	Chinese	English
GT Amateur	25.11	23.75
NVSB	**6.96**	**7.29**

4.2.2 Audio Quality and Vocal Tone Quality

To thoroughly evaluate our proposed model in audio quality and vocal tone quality, we compare subjective metric MOS-Q, MOS-V and objective metric MCD of audio samples generated by NVSB with the systems including: 1) *GT Mel*, amateur (A) and professional (P) version, where we first convert ground truth audio into mel-spectrograms, and then convert the mel-spectrograms back to audio using HiFi-GAN introduced in Section 4.1; 2) *Baseline*: the baseline model for SVB based on WaveNet with the number of parameters similar to *NSVB*, which adopts the same pitch correction method (SADTW) as *NSVB* does, and takes in the condition $\hat{c}_p$ defined in Section 3.3 to generate the mel-spectrogram optimized by the L_1 distance to x_p. MCD is calculated using the audio samples of *GT Mel P* as references.

The subjective and objective results on both Chinese and English datasets are shown in Table 3. We can see that 1) *NSVB* achieves promising results, with MOS-Q being less than those for ground truth professional recordings by only 0.1 and 0.12 on both datasets; 2) *NSVB* surpasses the *GT Mel A* in terms of MOS-V by a large margin, which indicates that *NSVB* successfully accomplishes the vocal tone improvement. 3) *NSVB* surpasses the baseline model on all the metrics distinctly, which proves the superiority of our proposed model; 4) *GT Mel P*, *NSVB* and *Baseline* all outperform *GT Mel A* in terms of MOS-V, which demonstrates that the proposed dataset PopBuTFy is reasonably labeled in respect of vocal tone.

4.3 Ablation Studies

We conduct some ablation studies to demonstrate the effectiveness of our proposed methods and some designs in our model, including latent-mapping, additional loss $\mathcal{L}_{map2}$ in the second training stage, and semi-supervised learning with extra unpaired, unlabeled data on Chinese songs.

Table 3: The Mean Opinion Score in audio quality (MOS-Q), vocal tone (MOS-V) with 95% confidence intervals and the Mean Cepstral Distortion (MCD) comparisons with ground-truth singing recordings and baseline model.

Method	MOS-Q	MOS-V	MCD
Chinese			
GT Mel P	4.21 ± 0.06	4.27 ± 0.10	-
GT Mel A	4.11 ± 0.07	3.51 ± 0.13	-
Baseline	3.90 ± 0.09	3.58 ± 0.18	7.609
NVSB	**4.11 ± 0.07**	**3.69 ± 0.17**	**7.068**
English			
GT Mel P	3.96 ± 0.11	3.96 ± 0.18	-
GT Mel A	3.67 ± 0.11	3.36 ± 0.19	-
Baseline	3.65 ± 0.12	3.37 ± 0.19	8.166
NVSB	**3.84 ± 0.06**	**3.63 ± 0.18**	**7.992**

4.3.1 Latent Mapping

We compare audio samples from NSVB with and without latent-mapping in terms of CMOS-V and MCD. From Table 4, we can see that the latent-mapping brings CMOS-V and MCD gains, which demonstrates the improvements in vocal tone from latent-mapping in our model. We visualize linear-spectrograms of *GT Mel A*, *GT Mel P*, *NSVB*, *NSVB w/o mapping* in Appendix B. The patterns of high-frequency parts in *NVSB* samples are comparatively similar to those in *GT Mel P* samples while *NSVB w/o mapping* sample resembles *GT Mel A* samples.

Table 4: The Comparison Mean Opinion Score in vocal tone (CMOS-V) and the Mean Ceptral Distortion (MCD) results of singing audio samples for latent mapping.

Method	CMOS-V	MCD
Chinese		
NVSB	0.000	**7.068**
NVSB w/o mapping	-0.100	7.069
English		
NVSB	0.000	**7.992**
NVSB w/o mapping	-0.330	8.115

4.3.2 Additional Loss $\mathcal{L}_{map2}$

As shown in Table 5, all the compared metrics show the effectiveness of $\mathcal{L}_{map2}$, which means that the additional loss $\mathcal{L}_{map2}$ is beneficial to optimizing the latent mapping function $\mathcal{M}$, working as a complement to the basic loss $\mathcal{L}_{map1}$.

Table 5: The Comparison Mean Opinion Score in audio quality (CMOS-Q), vocal tone (CMOS-V) and the Mean Ceptral Distortion (MCD) of singing audio samples.

Method	CMOS-Q	CMOS-V	MCD
Chinese			
NVSB	0.000	0.000	**7.068**
NVSB w/o $\mathcal{L}_{map2}$	-0.213	-0.760	7.237
English			
NVSB	0.000	0.000	**7.992**
NVSB w/o $\mathcal{L}_{map2}$	-0.060	-0.090	8.040

4.3.3 Semi-supervised Learning

To illustrate the advantage of the CVAE architecture that allows semi-supervised training, we compare NSVB trained with and without extra unpaired, unlabeled data on Chinese songs. The corresponding results are shown in Table 6. The compared metrics indicate the advantage of semi-supervised learning, which facilitates the learning of the latent representations for better sample reconstruction (audio quality) and better latent conversion (vocal tone quality).

Table 6: The Comparison Mean Opinion Score in audio quality (CMOS-Q), vocal tone (CMOS-V) and the Mean Ceptral Distortion (MCD) of singing audio samples.

Method	CMOS-Q	CMOS-V	MCD
NVSB	0.000	0.000	**7.068**
NVSB w/o extra data	-0.420	-0.070	7.283

5 Conclusion

In this work, we propose Neural Singing Voice Beautifier, the first generative model for the SVB task, which is based on a CVAE model allowing semi-supervised learning. For pitch correction, we propose a robust alignment algorithm: Shape-Aware Dynamic Time Warping (SADTW). For vocal tone improvement, we propose a latent mapping algorithm. To retain the vocal timbre during the vocal tone mapping, we also propose a new specialized SVB dataset named PopBuTFy containing parallel singing recordings of both amateur and professional versions. The experiments conducted on the dataset of Chinese and English songs show that NSVB accomplishes the SVB task (pitch correction and vocal tone improvement), and extensional ablation studies demonstrate the effectiveness of the proposed methods mentioned above.

Acknowledgments

This work was supported in part by Zhejiang Natural Science Foundation under Grant LR19F020006, National Natural Science Foundation of China under Grant No.61836002, No.62072397. Thank all the co-authors for their wonderful contributions, the enlightening opinions of the participants in the discussion, and the great efforts of singers and data annotators.

References

Taylor Berg-Kirkpatrick and Dan Klein. 2015. Gpu-friendly local regression for voice conversion. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1334–1338.

Merlijn Blaauw and Jordi Bonada. 2020. Sequence-to-sequence singing synthesis using the feed-forward transformer. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7229–7233. IEEE.

Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, and Ruoming Pang. 2020. Conformer: Convolution-augmented transformer for speech recognition. In *Interspeech 2020, 21st Annual Conference of the International Speech Communication Association, Virtual Event, Shanghai, China, 25-29 October 2020*, pages 5036–5040. ISCA.

Takuhiro Kaneko, Hirokazu Kameoka, Kou Tanaka, and Nobukatsu Hojo. 2019. Cyclegan-vc2: Improved cyclegan-based non-parallel voice conversion. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6820–6824. IEEE.

Diederik P. Kingma and Max Welling. 2014. Auto-encoding variational bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*.

Kazuhiro Kobayashi, Tomoki Toda, Graham Neubig, Sakriani Sakti, and Satoshi Nakamura. 2015. Statistical singing voice conversion based on direct waveform modification with global variance. In *Sixteenth Annual Conference of the International Speech Communication Association*.

Jungil Kong, Jaehyeon Kim, and Jaekyoung Bae. 2020. Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis. In *Advances in Neural Information Processing Systems*, volume 33, pages 17022–17033. Curran Associates, Inc.

Juheon Lee, Hyeong-Seok Choi, Chang-Bin Jeon, Junghyun Koo, and Kyogu Lee. 2019. Adversarially trained end-to-end korean singing voice synthesis system. *Proc. Interspeech 2019*, pages 2588–2592.

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

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

Songxiang Liu, Yuewen Cao, Dan Su, and Helen Meng. 2021b. Diffsvc: A diffusion probabilistic model for singing voice conversion.

Peiling Lu, Jie Wu, Jian Luan, Xu Tan, and Li Zhou. 2020. Xiaoicesing: A high-quality and integrated singing voice synthesis system. *Proc. Interspeech 2020*, pages 1306–1310.

Yin-Jyun Luo, Ming-Tso Chen, Tai-Shih Chi, and Li Su. 2018. Singing voice correction using canonical time warping. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 156–160. IEEE.

Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley. 2017. Least squares generative adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2794–2802.

Michael McAuliffe, Michaela Socolof, Sarah Mihuc, Michael Wagner, and Morgan Sonderegger. 2017. Montreal forced aligner: Trainable text-speech alignment using kaldi. In *Interspeech*, pages 498–502.

Greg Mori, Serge Belongie, and Jitendra Malik. 2005. Efficient shape matching using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(11):1832–1837.

Meinard Müller. 2007. Dynamic time warping. *Information retrieval for music and motion*, pages 69–84.

Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. 2016. Wavenet: A generative model for raw audio. In *9th ISCA Speech Synthesis Workshop*, pages 125–125.

Vadim Popov, Ivan Vovk, Vladimir Gogoryan, Tasnima Sadekova, Mikhail Kudinov, and Jiansheng Wei. 2021. Diffusion-based voice conversion with fast maximum likelihood sampling scheme. *arXiv preprint arXiv:2109.13821*.

Kaizhi Qian, Yang Zhang, Shiyu Chang, Xuesong Yang, and Mark Hasegawa-Johnson. 2019a. Autovc: Zero-shot voice style transfer with only autoencoder loss. In *International Conference on Machine Learning*, pages 5210–5219. PMLR.

Kaizhi Qian, Yang Zhang, Shiyu Chang, Xuesong Yang, and Mark Hasegawa-Johnson. 2019b. AutoVC: Zero-shot voice style transfer with only autoencoder loss. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 5210–5219. PMLR.

Yi Ren, Xu Tan, Tao Qin, Jian Luan, Zhou Zhao, and Tie-Yan Liu. 2020. Deepsinger: Singing voice synthesis with data mined from the web. *arXiv preprint arXiv:2007.04590*.

Dario Rethage, Jordi Pons, and Xavier Serra. 2018. A wavenet for speech denoising. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5069–5073. IEEE.

Sebastian Rosenzweig, Simon Schwär, Jonathan Driedger, and Meinard Müller. 2021. Adaptive pitch-shifting with applications to intonation adjustment in a cappella recordings. In *Proceedings of the International Conference on Digital Audio Effects (DAFx), Vienna, Austria*.

Joan Serrà, Santiago Pascual, and Carlos Segura Perales. 2019. Blow: a single-scale hyperconditioned flow for non-parallel raw-audio voice conversion. *Advances in Neural Information Processing Systems*, 32:6793–6803.

Berrak Sisman and Haizhou Li. 2020. Generative adversarial networks for singing voice conversion with and without parallel data. In *Speaker Odyssey*, pages 238–244.

Berrak Sisman, Karthika Vijayan, Minghui Dong, and Haizhou Li. 2019. Singan: Singing voice conversion with generative adversarial networks. In *2019 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pages 112–118. IEEE.

Kihyuk Sohn, Honglak Lee, and Xinchen Yan. 2015. Learning structured output representation using deep conditional generative models. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 3483–3491.

David Sundermann and Hermann Ney. 2003. Vtln-based voice conversion. In *Proceedings of the 3rd IEEE International Symposium on Signal Processing and Information Technology (IEEE Cat. No. 03EX795)*, pages 556–559. IEEE.

Fernando Villavicencio and Jordi Bonada. 2010. Applying voice conversion to concatenative singing-voice synthesis. In *Eleventh annual conference of the international speech communication association*.

Yusuke Wada, Yoshiaki Bando, Eita Nakamura, Katsutoshi Itoyama, and Kazuyoshi Yoshii. 2017. An adaptive karaoke system that plays accompaniment parts of music audio signals synchronously with users'singing voices. In *SMC*, pages 110–116.

Sanna Wager, George Tzanetakis, Cheng-i Wang, and Minje Kim. 2020. Deep autotuner: A pitch correcting network for singing performances. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 246–250. IEEE.

Li Wan, Quan Wang, Alan Papir, and Ignacio Lopez Moreno. 2018. Generalized end-to-end loss for speaker verification. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4879–4883. IEEE.

Chao Wang, Zhonghao Li, Benlai Tang, Xiang Yin, Yuan Wan, Yibiao Yu, and Zejun Ma. 2021a. Towards high-fidelity singing voice conversion with acoustic reference and contrastive predictive coding.

Disong Wang, Liqun Deng, Yu Ting Yeung, Xiao Chen, Xunying Liu, and Helen Meng. 2021b. Vqmivc: Vector quantization and mutual information-based unsupervised speech representation disentanglement for one-shot voice conversion. *arXiv preprint arXiv:2106.10132*.

Jie Wu and Jian Luan. 2020. Adversarially trained multi-singer sequence-to-sequence singing synthesizer. *arXiv preprint arXiv:2006.10317*.

Hui Bu Yao Shi, Shaoji Zhang Xin Xu, and Ming Li. 2020. Aishell-3: A multi-speaker mandarin tts corpus and the baselines.

Sangeon Yong and Juhan Nam. 2018. Singing expression transfer from one voice to another for a given song. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 151–155. IEEE.

Siyang Yuan, Pengyu Cheng, Ruiyi Zhang, Weituo Hao, Zhe Gan, and Lawrence Carin. 2020. Improving zero-shot voice style transfer via disentangled representation learning. In *International Conference on Learning Representations*.

Heiga Zen, Viet Dang, Rob Clark, Yu Zhang, Ron J. Weiss, Ye Jia, Zhifeng Chen, and Yonghui Wu. 2019. Libritts: A corpus derived from librispeech for text-to-speech. In *Interspeech 2019, 20th Annual Conference of the International Speech Communication Association, Graz, Austria, 15-19 September 2019*, pages 1526–1530. ISCA.

Liqiang Zhang, Chengzhu Yu, Heng Lu, Chao Weng, Chunlei Zhang, Yusong Wu, Xiang Xie, Zijin Li, and Dong Yu. 2020. Durian-sc: Duration informed attention network based singing voice conversion system. In *Interspeech 2020, 21st Annual Conference of the International Speech Communication Association, Virtual Event, Shanghai, China, 25-29 October 2020*, pages 1231–1235. ISCA.

Yi Zhao, Wen-Chin Huang, Xiaohai Tian, Junichi Yamagishi, Rohan Kumar Das, Tomi Kinnunen, Zhenhua Ling, and Tomoki Toda. 2020. Voice conversion challenge 2020: Intra-lingual semi-parallel and cross-lingual voice conversion. *arXiv preprint arXiv:2008.12527*.

Feng Zhou and Fernando De la Torre. 2012. Generalized time warping for multi-modal alignment of human motion. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1282–1289. IEEE.

Feng Zhou and Fernando Torre. 2009. Canonical time warping for alignment of human behavior. In *Advances in Neural Information Processing Systems*, volume 22. Curran Associates, Inc.

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232.

Xiaobin Zhuang, Huiran Yu, Weifeng Zhao, Tao Jiang, Peng Hu, Simon Lui, and Wenjiang Zhou. 2021. Karatuner: Towards end to end natural pitch correction for singing voice in karaoke.

A Details of Model Structure

The details of the adversarial discriminator, the content encoder, and WaveNet structure are shown in Figure 5, Figure 8, and Figure 6. The hidden size of CVAE model, latent variable and discriminator are 256, 128 and 128 respectively. We train NSVB on a single V100 32G GPU for almost 22 hours to finish two-stage training.

A.1 Multi-window Discriminator

As shown in Figure 5, our multi-window discriminator consists of 2 unconditional discriminator parts with fixed window sizes. Each unconditional discriminator contains N layers of Conv units. In our model, we set $N = 3$. The Conv units are all 1-D convolutional networks with ReLU activation and spectral normalization. The outputs of these unconditional discriminators are then concatenated and linearly projected to form the output.

A.2 WaveNet

As shown in Figure 6, the WaveNet unit used in the VAE encoder and decoder of NVSB consists of a 1D convolution layer with ReLU to preprocess the input, and a group of sub-layers with residual connection between adjacent ones. Each sub-layer contains a 1×1 convolutional layer to process the input condition and a 3×3 convolutional layer for residual input. After that, they got fused by being added up, then processed by tanh and sigmoid separately and then multiplied together. Finally, they produce a residual output for the next sub-layer and a skip-out. Lastly, two layers of 1D convolution and a ReLU process the summed skip-out to produce output.

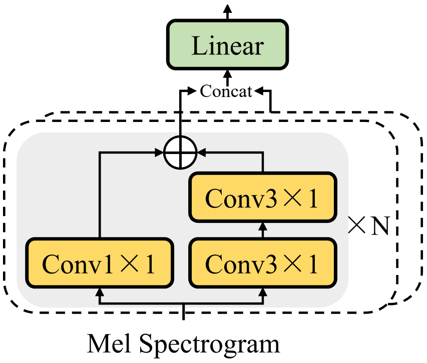


Figure 5: Multi-window discriminator structure used in NSVB

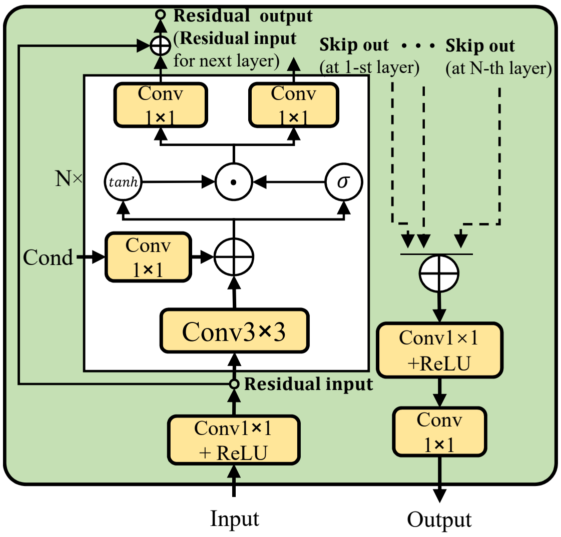
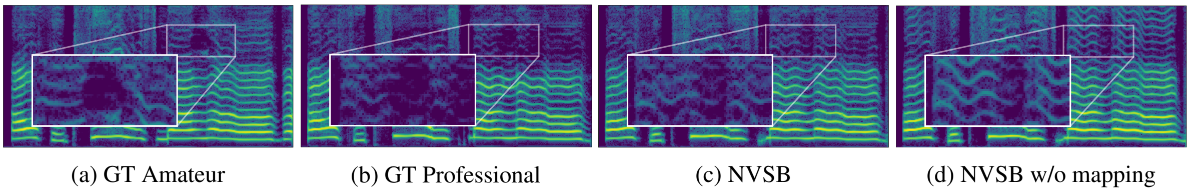


Figure 6: WaveNet structure used in NSVB

A.3 Content Encoder

As shown in Figure 8, the content encoder is the combination of several conformer encoder layers in pink rectangle along with a 3-layer prenet. The kernel size of the convolutional layer for prenet is 5. The hidden size is 256. We use 4 heads in the multi-head self-attention part. And we use 31 stacked conformer encoder layers to form this module. During pre-training, an ASR transformer decoder is attached to decode texts out for regular ASR training. After pre-training, only the encoder and the prenet part is used to extract PPG features



(a) GT Amateur

(b) GT Professional

(c) NVSB

(d) NVSB w/o mapping

Figure 7: Linear-spectrogram visualizations for the ablation study on latent mapping.

from mel-spectrograms of audio samples.

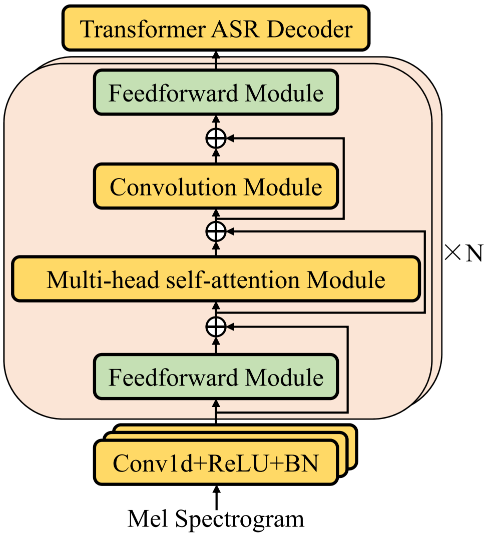


Figure 8: Content encoder used in NSVB

B Linear-spectrograms Visualizations

We visualize four linear-spectrograms generated with the same content. It seems that the professional vocal tone is related to certain patterns in the high-frequency region of the spectrograms. In the future, SVB may be accomplished in a more fine-grained way together with the knowledge in vocal music.

C Details in subjective evaluations

During testing, each audio sample is listened to by at least 10 qualified testers, all majoring in vocal music. We tell all testers to focus on one aspect and ignore the other aspect when scoring MOS/CMOS of each aspect. For MOS, each tester is asked to evaluate the subjective naturalness of a sentence on a 1-5 Likert scale. For CMOS, listeners are asked to compare pairs of audio generated by systems A and B and indicate which of the two audio they prefer and choose one of the following scores: 0 indicating no difference, 1 indicating small difference, 2 indicating a large difference. For audio quality evaluation (MOS-Q and CMOS-Q), we tell listeners to "focus on examining the naturalness, pronunciation and sound quality, and ignore the differences of singing vocal tone". For vocal tone evaluations (MOS-V and CMOS-V), we tell listeners to "focus on examining singing vocal tone of the song, and ignore the differences of audio quality (e.g., environmental noise, timbre)". We split evaluations for main experiments and ablation studies into several groups for them. They are asked to take a break for 15 minutes between each group of experiments to remain focused during subjective evaluations. All testers get reasonably paid.

D Limitations and Solutions

SADTW is a kind of advanced APC method, which is designed for fine-tuning the amateur recording, but not for the case when the amateur recordings are completely out of tune. In the latter case, we recommend people to use Singing Voice Synthesis (synthesizing waveform from PPG and MIDI) + Singing Voice Conversion (converting the vocal timbre of the synthesized waveform into the user's), or some Speech to Singing (STS) methods. In addition, SADTW provides a score representing the similarity of two pitch curves, which could be used to determine what kind of SVB solution should be chosen.

E Potential Negative Societal Impacts

This work develops a possible automatic way for singing voice beautification, which may cause unemployment for people with related occupations. In addition, there is the potential for harm from piracy and abuse of our released recordings. Thus, we choose the dataset license: CC by-nc-sa 4.0.

F Ethics Statement

F.1 For every submission

F.1.1 Did you discuss the *limitations* of your work?

Yes. Appendix D.

F.1.2 Did you discuss any potential *risks* of your work?

Yes. Appendix E.

F.1.3 Do the abstract and introduction summarize the paper's main claims?

Yes. Abstract.

F.2 Did you use or create *scientific artifacts*?

Yes.

If yes:

F.2.1 Did you cite the creators of artifacts you used?

Yes. Section 4.1.

F.2.2 Did you discuss the *license or terms* for use and/or distribution of any artifacts?

Yes. Section 4.1 and Appendix E.

F.2.3 Did you discuss if your use of existing artifact(s) was consistent with their *intended use*, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?

Yes. Appendix E.

F.2.4 Did you discuss the steps taken to check whether the data that was collected/used contains any *information that names or uniquely identifies individual people* or *offensive content*, and the steps taken to protect / anonymize it?

Yes. The participants are asked to use bogus names in the data collection process, which means we do not know who sang these recordings.

F.2.5 Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?

Yes. Section 4.1.

F.2.6 Did you report relevant statistics like the number of examples, details of train/test/dev splits, etc. for the data that you used/created?

Yes. Section 4.1.

F.3 Did you run *computational experiments*?

Yes.

If yes:

F.3.1 Did you report the *number of parameters* in the models used, the *total computational budget* (e.g., GPU hours), and *computing infrastructure* used?

Yes. Section 4.1.

F.3.2 Did you discuss the experimental setup, including *hyperparameter search* and *best-found hyperparameter* values?

Yes. Section 4.1.

F.3.3 Did you report *descriptive statistics* about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?

Yes. Section 4.2.

F.3.4 If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?

Yes. Section 4.1.

F.4 Did you use *human annotators* (e.g., crowdworkers) or *research with human subjects*?

If you answer **Yes**, provide the section number; if you answer **No**, you can skip the rest of this section.

Yes.

If yes:

F.4.1 Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?

Yes. Appendix C.

F.4.2 Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such *payment is adequate* given the participants' demographic (e.g., country of residence)?

No. Trade secret.

F.4.3 Did you discuss whether and how *consent* was obtained from people whose data you're using/curating (e.g., did your instructions explain how the data would be used)?

Yes. Section 4.1 and our code project.

F.4.4 Was the data collection protocol *approved (or determined exempt)* by an ethics review board?

N/A.

F.4.5 Did you report the basic demographic and geographic characteristics of the *annotator* population that is the source of the data?

No. They're completely anonymous, and we don't know.