SinTechSVS: A Singing Technique Controllable Singing Voice Synthesis System

Junchuan Zhao, Low Qi Hong Chetwin, Ye Wang, *Member, IEEE*

***Abstract*— The precise control of singing techniques is of utmost importance in achieving emotionally expressive vocal performances. To bridge the gap between current Singing Voice Synthesis (SVS) systems and human singers, our paper focuses on developing an SVS system that allows for control over singing techniques. In this paper, we introduce SinTechSVS, a singing technique controllable SVS system composed of a singing technique annotator, a singing technique controllable synthesizer, and a singing technique recommender. Our approach leverages transfer learning for efficient singing technique annotation and adapts the DiffSinger framework with additional style encoders and an attention-based singing technique local score (STLS) module to enhance singing technique controllability. We also propose a Seq2Seq singing technique recommender for the new task of Singing Technique Recommendation (STR). Experimental results demonstrate that SinTechSVS significantly improves the quality and expressiveness of synthesized vocal performances, with comparable general synthesis capabilities to state-of-the-art SVS systems and enhanced control over singing techniques, as evidenced by objective and subjective evaluations. To the best of our knowledge, SinTechSVS is the first SVS capable of controlling singing techniques.**

***Index Terms*—Singing voice synthesis, singing voice synthesis conditioned on singing techniques, singing technique classification, singing technique recommendation, metric, deep learning.**

I. Introduction

SINGING voice synthesis (SVS) is the task of synthesizing an expressive singing voice for a given music score and corresponding lyrics by using computing models [1]. In the past, singing voice synthesis (SVS) systems mostly adopted concatenative [2], [3], [4], [5], [6], [7], [8], [9] and Hidden Markov Model (HMM) [10], [11], [12], [13], [14] approaches. The field of singing voice synthesis (SVS) has witnessed significant advancements in recent years due to the development of advanced deep learning methods. These methods have enabled the creation of highly sophisticated SVS systems that can generate singing voices with high accuracy in terms of pronunciation, pitch, and duration [15], [16], [17]. However, it is worth noting that professional singers rely on more than just the accurate delivery of notes and lyrics. Emotional expressiveness and delivery of performances are also crucial aspects that make for a great singer [18], [19]. Consequently, the development of flexible expression controls has become a crucial area of focus in the latest SVS systems.

Singing voice synthesis with expression control is the task of conditioning voice-related features while synthesizing the singing voice [18]. In the context of SVS with expression control, there have been studies that focused on intensity [20], vibrato [21], [22], [14], timbre [23], [14], breath [20], singer identity [23], [24], and emotion [21] as control signals. However, despite the significant strides made by deep learning-based SVS systems in generating natural and realistic singing voices, research on singing voice synthesis with singing technique control is still lacking, as suggested by Cho et al. [1]. Real singers convey expression through deliberate choice of singing techniques, and thus singing techniques should be the primary focus of SVS research.

As highlighted by [25], the concept of singing technique encompasses the utilization of extended techniques in human singing, serving as a crucial component in vocal performances. This paper mainly explores singing techniques within the POP genre as discussed by [25] since it includes comprehensive research in various singing techniques outlined within this genre and the availability of abundant datasets for this kind of research. Although there are additional singing techniques, such as 'staccato' and 'legato forte,' listed in other genres, they are not the primary focus of our study. This decision is due to the absence of reference definitions and datasets pertaining to these particular techniques.

Unfortunately, the establishment of robust singing technique control encounters several formidable challenges within the realm of research and development: (1) The dearth of comprehensive and meaningful annotations pertaining to singing techniques presents a significant obstacle. The absence of such annotations hinders the capacity to train and fine-tune SVS systems effectively. (2) Addressing the need to condition SVS systems for the synthesis of audio with specific singing techniques is another intricate challenge. Achieving this conditioning necessitates innovative approaches to incorporate singing technique nuances into the synthesis process. (3) Evaluating the efficacy and controllability of SVS systems in replicating singing techniques represents yet another intricate challenge. Accurately assessing the performance of these systems in capturing and reproducing nuanced vocal expressions demands the development of robust evaluation methodologies.

This paper aims to establish a deep learning-based SVS system that allows users to control the singing techniques of the synthesized singing voices. The contributions are summarized as follows:

- We introduce SinTechSVS, a singing technique control-

Junchuan Zhao, Low Qi Hong Chetwin, and Ye Wang, are affiliated with the School of Computing, National University of Singapore, Singapore. Ye Wang is the correspondence author of this paper. This paper has supplementary downloadable material available at http://ieeexplore.ieee.org, provided by the author. The material includes the manual singing technique annotations for Opencpop and the demos of SinTechSVS. Contact junchuan@comp.nus.edu.sg for further questions about this work.

lable SVS system comprising three main modules: the singing technique annotator (STA), the singing technique controllable singing voice synthesizer (SVS), and the singing technique recommender (STR). Additionally, we enhance the SVS model with singing technique control by integrating an attention-based singing technique local score module (STLS) to improve controllability within the Diffusion-based acoustic model. To fully leverage the capabilities of SinTechSVS, we propose a transformer-based singing technique recommender capable of generating utterance-level singing technique sequences from music scores. This automates the process of determining singing technique labels for entire song utterances, simplifying user interaction and reducing complexity.

- We propose a data-efficient singing technique annotation framework using transfer learning and put forward a new singing technique classification model with a Temporal Pyramid Pooling (TPP) layer. This annotation framework effectively addresses the challenge posed by the scarcity of high-quality, publicly available annotated singing voice datasets.
- We present two inspired evaluation metrics, Style Reclassification Accuracy (SR-Acc) and Style Match Rate (SMR), to measure the controllability of singing techniques, both subjectively and objectively. Experimental results demonstrate the effectiveness of SinTechSVS in both unconditional and conditional synthesis capability.

II. Related Works

A. Singing Voice Synthesis Based on Deep Learning

In the later part of the 2000s and early 2010s, Singing Voice Synthesis (SVS) experienced widespread popularity, primarily through concatenation-based systems, such as VOCALOID [26], [27], and Hidden Markov Model (HMM)-based systems, such as the one introduced by Saino et al. [10], which concurrently modeled lyrics, tones, and durations. Nonetheless, within the last few years, deep learning has emerged, leading to numerous breakthroughs and enabling SVS to achieve unparalleled levels of naturalness and accuracy. Various concatenation-based and HMM-based SVS systems have also transitioned to deep neural networks (DNNs), such as Sinsy, which assumes that DNNs surmount the over-smoothing that occurs within HMM models [28]. Comparatively, SVS systems founded on neural networks, as opposed to HMM-based ones, typically achieve better results with respect to the quality and naturalness of synthesized audio signals [29], [30], [16], [31]. Cho et al. [1] conducted a survey that summarized five contemporary, state-of-the-art Deep-learning-based (DL-based) SVS systems that employed various neural networks. DiffSinger, which applies a Diffusion-based acoustic model, [30], showed very promising qualitative advancements compared to other state-of-the-art models, in both SVS and text-to-speech (TTS) tasks. In light of their findings, we propose a Diffusion-based singing technique controllable acoustic model. The authors of the survey also underscored that future SVS systems will require greater data efficiency, and expression controls that activate heightened variability in emotions and singing techniques, in addition to more open datasets and interpretable models. These challenges served as primary motivations for our work and the contributions we have made towards achieving data efficiency and singing technique control within our proposed SVS system.

B. Expression Control in Singing Voice Synthesis

Umbert et al. [18] conducted a detailed investigation of expression control and performance modeling, which encompassed a comprehensive inventory and classification of a multitude of potentially valuable features that could be manipulated. They also identified various existing approaches to expression control that differed in parameters such as timbre, formants, and vibrato. However, the authors pointed out that it would be a novel challenge to achieve intricate, dynamic, and expressive modifications in singing. These modifications relate to emotions, singing styles, and techniques.

Recently, the development of deep learning has expanded the range of controllable expression features in SVS.

Singer-ID control, which enables users to specify a target singer while synthesizing singing audio. This has received considerable attention due to the availability of rich annotations and the easily distinguishable nature of singer-IDs. Researchers have made relevant contributions to this field [21], [32], [33], [23], [22], [24].

Vibrato control is another significant control signal that involves the ability to control the presence and characteristics of 'Vibrato' in the synthesized singing voice. Liu et al. [32] proposed a deep learning-based SVS system capable of controlling 'Vibrato' patterns in synthesized singing voices. Song et al. [22] proposed a DL-based vibrato model, designed to enhance singing naturalness by controlling multiple aspects of vibrato. Additionally, Sinsy [28] introduced innovative methods for modeling pitch and vibrato, elevating the expressiveness of singing voices.

Emotion control refers to the ability to control the emotional content of the synthesized singing voice. Kim et al. [21] proposed U-Singer, which is the first deep learning-based multi-singer emotional SVS capable of controlling emotional intensity by regulating the subtle fluctuations in pitch, energy, and duration of phonemes while synthesizing voices.

Singing Technique control enables users to specify a desired technique for synthesized singing voices. Little empirical research exists on precise control within Singing Voice Synthesis (SVS). Lee et al. [20] explored intensity control for energy and timbre variations, but their study lacked annotations and comprehensive evaluation. In the context of singing voice conversion and editing, there are a few works that discuss vocal technique control. [34] presents a system capable of transferring vocal expressions that involve variations and fluctuations in fundamental frequency, such as 'vibrato', 'kobushi', and 'glissando'. While it can control specific singing techniques by simply specifying their type, it is limited to pitch-based techniques. Conversely, [35] proposes a method to synthesize a singing voice by emulating the timbre changes of a user's singing voice. This method is controlled based on the singer's database rather than the type

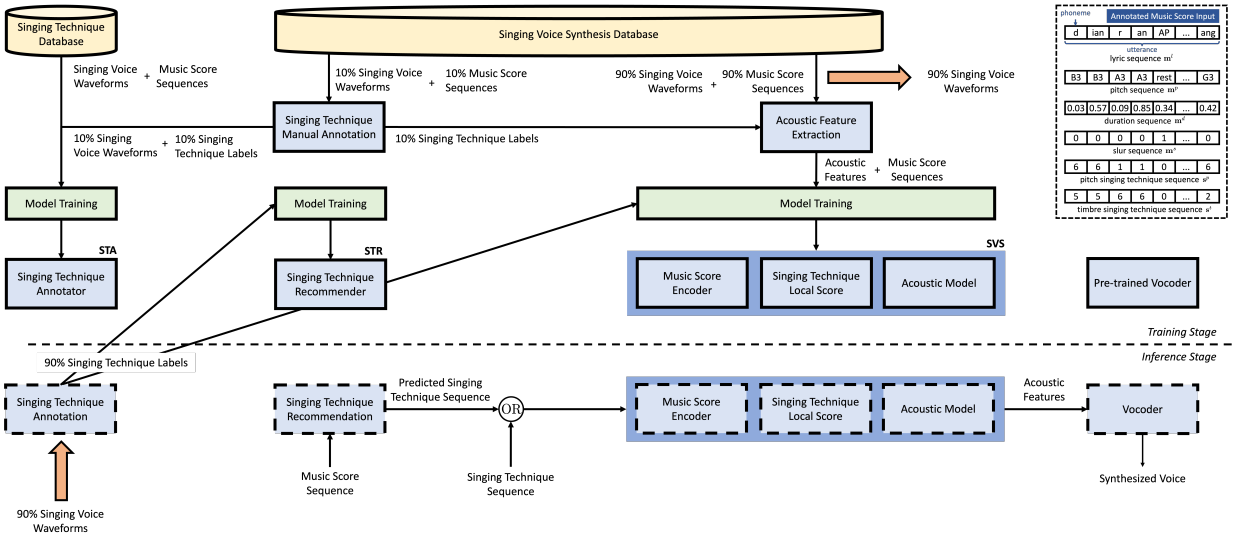


Fig. 1: The architecture of our proposed SinTechSVS system. It consists of three key components: singing technique annotator (STA), singing voice synthesizer conditioned on singing techniques (SVS), and singing technique recommender (STR). The 'OR' symbol in this figure means that the input of SVS is either by a user-input singing technique sequence or the predicted singing technique sequence from the singing technique recommender.

of singing technique, and it exclusively includes timbre-based techniques. As a result, we aim to contribute a general solution with a reference value for singing technique control in SVS. Our aim is to incorporate both pitch-based and timbre-based singing techniques, allowing for the control of singing styles by specifying their respective types.

C. Singing Techniques on Music Information Retrieval (MIR) research

Singing techniques play a crucial role in musical and emotional expression. Prior research, such as the Phonation Mode framework [36] categorizes singing into phonation modes like 'Breathy', 'Neutral', 'Flow', and 'Pressed'. The VocalSet dataset [37] classifies techniques within scales, arpeggios, long tones, and excerpts, but its focus on opera-style singing and varying definitions made it less relevant to our study on commercial popular music. Datasets like KVT [38] and MVD [39] capture singing techniques in commercial music contexts. Yet, KVT prioritizes emotional expression labels over technique definitions, while MVD concentrates on extreme vocal techniques in heavy metal music, diverging from our popular music context.

Yamamoto et al.'s work [25] stands out for curating the COSIAN dataset from J-POP songs, offering precise definitions and clear descriptions of 17 distinct techniques categorized as 'Timbre' and 'Pitch.' This dataset formed the basis for our research on singing technique control in commercial popular music.

We thoroughly analyzed Yamamoto's research and COSIAN dataset, aiming to extract insights into singing techniques and their relationships. Notably, our analysis revealed significant time overlaps between annotations of different singing technique types (e.g., timbral, pitch), suggesting their simultaneous utilization. Also, we noticed that it's rare for the same singing technique to occur simultaneously. Therefore, we opted to categorize the singing techniques into two distinct groups, pitch and timbral, to facilitate independent control over both parameters and allow for their joint usage in singing. We also added the 'Belting' technique [40], which is commonly used to convey intense emotions in climactic parts of songs. The 'Miscellaneous' category was disregarded as it was deemed irrelevant to singing. Due to the abundance of existing literature and research on 'Vibrato' in both classification [41] and SVS tasks [28], [22], we decided to exclude it. Vibrato was excluded due to its abundance in previous research in SVS [28], [22]. At the same time, 'Whisper' and 'Hiccup' techniques were removed because they had a negligible amount of labels in comparison to the other classes, as illustrated in Table III.

Table I shows the singing techniques that we focused on and their corresponding descriptions defined in Yamamoto et al.'s [25] paper, with the exception of 'Belting' which was defined by us. 'Straight' pitch and 'Regular' timbre refer to the absence of any other singing technique listed in Table I.

III. Method

A. Model Architecture

The overall architecture of the proposed model is shown in Fig. 1. The proposed model is composed of three key components: a singing technique annotator (STA), a singing technique synthesizer conditioned on singing techniques (SVS), and a singing technique recommender (STR). The details of these components are described as follows.

Singing Technique Annotator (STA): The singing technique annotator annotates the singing techniques on the original SVS dataset $\mathcal{M} = \{\mathbf{m}^p, \mathbf{m}^l, \mathbf{m}^d, \mathbf{m}^s, \mathbf{a}\}$, where $\mathbf{m}^p$, $\mathbf{m}^l$, $\mathbf{m}^d$, $\mathbf{m}^s$ respectively represents the phoneme-level pitch sequence, the lyrics sequence, the duration sequence, and the slur

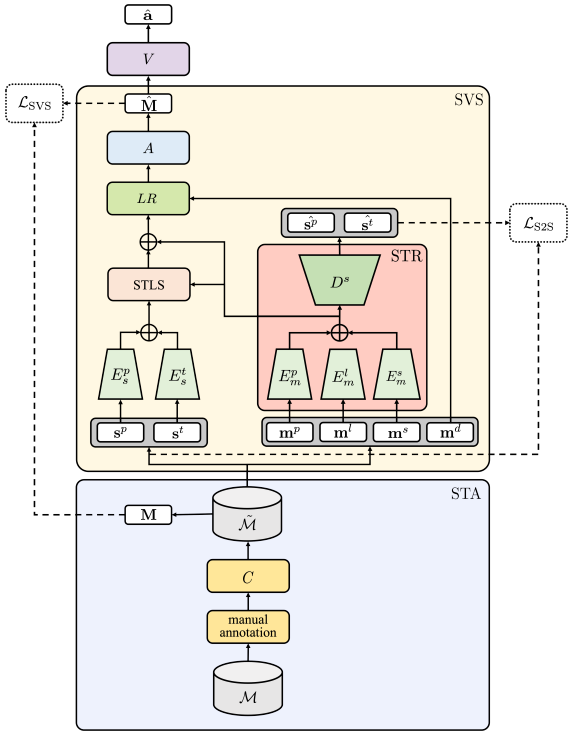


Fig. 2: The detailed architecture of SinTechSVS with symbol representations.

sequence, $\mathbf{a}$ represents the singing audio signal. More specifically, $\mathbf{m}^p$ represents the sequence of pitch symbols (C4, D4, ...), while $\mathbf{m}^l$ denotes the sequence of phonemes. Ultimately, the annotated SVS dataset $\tilde{\mathcal{M}} = \{\mathbf{m}^p, \mathbf{m}^l, \mathbf{m}^d, \mathbf{m}^s, \mathbf{s}^p, \mathbf{s}^t, \mathbf{a}\}$ is obtained, where $\mathbf{s}^p$ and $\mathbf{s}^t$ respectively represents the pitch singing technique sequence and the timbral singing technique sequence. The singing technique sequences correspond to phoneme-level sequences of singing technique labels. Therefore, the lengths of $\mathbf{m}^p, \mathbf{m}^l, \mathbf{m}^d, \mathbf{m}^s, \mathbf{s}^p$, and $\mathbf{s}^t$ are identical. The STA module contains two parts: the manual annotator, which annotates 10% of the original SVS dataset $\mathcal{M}$ through meticulous human labeling for precision and reliability; and the singing technique classifier C, annotating the remaining 90% of the dataset automatically. The architectural specifics of the singing technique classifier are expounded upon in III-B. The procedures for constructing the annotated dataset $\tilde{\mathcal{M}}$ are outlined in III-E, while IV-A describes it in full detail.

TABLE I: The description of each singing technique.

Technique	Type	Description
Falsetto	Timbre	Sung by falsetto register.
Breathy	Timbre	Sung by breathy sound.
Vocal Fry	Timbre	Sung by a creaky voice and pulse register phonation.
Belting	Timbre	Sung by a yell-like, powerful voice.
Scooping	Pitch	An upper continuous pitch change.
Drop	Pitch	A lower continuous pitch change.
Bend	Pitch	A short tremolo or U/inverted-U shaped pitch change.
Melisma	Pitch	A musical arrangement in which several notes are applied to one syllable of a lyric.

Singing Voice Synthesizer Conditioned on Singing Techniques (SVS): The singing voice synthesizer processes the annotated SVS dataset $\tilde{\mathcal{M}}$ to produce synthesized audio $\hat{\mathbf{a}}$, following a standard deep learning architecture structure described in literature [1]. This architecture includes an encoder E, an acoustic model A, and a vocoder V, consistent with state-of-the-art SVS systems [30], [16], [17], [15], [42], [43] The encoder architectures, follow the structure of DeepSinger [15], consist of specific encoders: E^p_m encodes pitch sequence $\mathbf{m}^p$, E^l_m encodes lyrics sequence $\mathbf{m}^l$, and E^s_m encodes slur sequence $\mathbf{m}^s$, transforming the sequences into embeddings, respectively $\mathbf{e}^p_m$, $\mathbf{e}^l_m$, and $\mathbf{e}^s_m$. To integrate singing techniques and improve control, we introduce singing technique encoders and a Singing Technique Local Score (STLS) module. The pitch and timbral singing technique encoders convert $\mathbf{s}^p$ and $\mathbf{s}^t$ respectively into embedding sequences $\mathbf{e}^p_s$ and $\mathbf{e}^t_s$, following the structure of the pitch encoder E^p_m, which includes an embedding layer and several Transformer Blocks [15]. The music score embedding sequence $\mathbf{e}_m$ is derived by combining $\mathbf{e}^p_m$, $\mathbf{e}^l_m$, $\mathbf{e}^s_m$, and the singing technique embedding sequence $\mathbf{e}_s$ formed by $\mathbf{e}^p_s$ and $\mathbf{e}^t_s$. The STLS module processes the embedding sequences ($\mathbf{e}_m$, $\mathbf{e}_s$) into a singing technique local score vector $\mathbf{e}_f$. These embedding sequences ($\mathbf{e}_m$ and $\mathbf{e}_f$) are combined and extended in length using a duration sequence $\mathbf{m}^d$ and a length regulator LR to match the spectrogram sequences. The resulting music condition sequence $\mathbf{e}_c$ serves as input to the acoustic model A. Further details on the STLS module's architecture can be found in Section III-C. Details regarding the architecture of the STLS module can be found in Section III-C. Subsequently, the acoustic model A takes $\mathbf{e}_c$ as input and generates the mel-spectrogram $\hat{\mathbf{M}}$. This mel-spectrogram is then processed by the vocoder V to produce the synthesized audio $\hat{\mathbf{a}}$. We use a pre-trained HiFi-GAN model [29] for the vocoder, which is designed for high-fidelity speech and singing voice synthesis[1]. While there may be better alternatives, it is worth noting that HiFi-GAN was trained on the same dataset as ours, addressing potential issues with imprecise pitch control. Additionally, for fair evaluation, we chose to use the same vocoder as our comparison models. Enhancing the vocoder's capabilities could potentially boost synthesis performance, which we intend to explore in future work. The overall SVS process can be formulated as displayed in (1):

$$\begin{aligned} \mathbf{e}_m &= \mathbf{e}^p_m + \mathbf{e}^l_m + \mathbf{e}^s_m, \mathbf{e}_s = \mathbf{e}^p_s + \mathbf{e}^t_s \\ \mathbf{e}_f &= \mathrm{STLS}(\mathbf{e}_m, \mathbf{e}_s) \\ \mathbf{e}_c &= LR(\mathbf{e}_m + \mathbf{e}_f, \mathbf{m}^d) \\ \hat{\mathbf{a}} &= V(A(\mathbf{e}_c)) \end{aligned} \tag{1}$$

Singing Technique Recommender (STR): The STR system predicts pitch and timbral singing technique sequences, denoted as $\hat{\mathbf{s}^p}$ and $\hat{\mathbf{s}^t}$, based on the pitch, lyrics, and slur sequences $\mathbf{m}^p$, $\mathbf{m}^l$, and $\mathbf{m}^s$. The STR reuses the pitch, lyric,

[1]https://github.com/MoonInTheRiver/DiffSinger/releases/download/pretrain-model/0109_hifigan_bigpopcs_hop128.zip

and slur encoders E_m^p, E_m^l, and E_m^s from the SVS system to obtain the corresponding embedding sequences $\mathbf{e}_m^p$, $\mathbf{e}_m^l$, and $\mathbf{e}_m^s$. These embeddings are then combined to form the music score embedding sequence $\mathbf{e}_m$ as shown in Eq. 1. Subsequently, $\mathbf{e}_m$ is fed into the singing technique decoder D^s, consisting of two location-aware attention-based GRU units [44], to decode $\hat{\mathbf{s}^p}$ and $\hat{\mathbf{s}^t}$ in an autoregressive manner. The process of STR can be expressed as shown in (2):

$$\hat{\mathbf{s}^p}, \hat{\mathbf{s}^t} = D^s(\mathbf{e}_m) \tag{2}$$

B. Singing Technique Classifier

Building upon the models in subsection III-A, this section describes the implementation of the annotator mentioned in the pipeline in full detail.

In recent years, CNNs have dominated audio classification and recognition tasks, handling audio waveforms represented as mel-spectrograms or MFCCs. Previous studies have shown CNNs' effectiveness, leveraging Transfer Learning with networks like VGG19 or custom deep CNNs [45], [25], [46], [47]. Similarly, we propose a deep CNN model for singing technique classification. The singing technique classifier (STC) takes word-level mel-spectrograms $\mathbf{M}_w$ as input, comprising a feature extractor F^s and an output head h. It predicts word-level timbre and pitch techniques, denoted by $\mathbf{w}^t$ and $\mathbf{w}^p$, respectively. The formulation of the STC is as follows.

$$\mathbf{w}^p, \mathbf{w}^t = h(f(\mathbf{M}_w)) \tag{3}$$

A detailed breakdown of our classifier's parameters can be seen in Table II. Our classifier has the following key customization: (1) Temporal Pyramid Pooling [48] to handle variable-sized inputs; (2) Transfer Learning to compensate for the lack of data. We employ the Temporal Pyramid Pooling (TPP) layer, derived from He et al.'s Spatial Pyramid Pooling (SPP) layer [49], to convert variable-length audio segments into fixed-length feature vectors. The TPP layer pools input solely along the time dimension. Additionally,

TABLE II: The parameters of the CNN-based singing technique classifier. BN=Batch Normalization, B=Batchsize, W=Width of the output feature map. The input size of the singing technique classifier (STC) is B × C × W × H, C = 2 since the audios are in stereo format; W is the width of the mel-spectrograms; H is the height of the mel-spectrograms.

Module	Parameters	Output Size
Feature Extractor	Conv(2,32,(3×3))+BN+ReLU	B×32×64×W
	MaxPool((2,1))	B×32×32×W
	Conv(32,64,(5×5))+BN+ReLU	B×64×32×W
	MaxPool((2,1))	B×64×16×W
	Conv(64,128,(3×3))+BN+ReLU	B×128×16×W
	MaxPool((2,1))	B×128×8×W
	Conv(128,128,(5×5))+BN+ReLU	B×128×8×W
	MaxPool((2,1))	B×128×4×W
	AvgPool((2,1))	B×128×2×W
	TPP((1,2,4))	B×128×14
	FC1(1792,64)+BN+ReLU	B×64
Output Head	FC2(64,10)	B×10
	Softmax(5), Softmax(5)	B×5, B×5

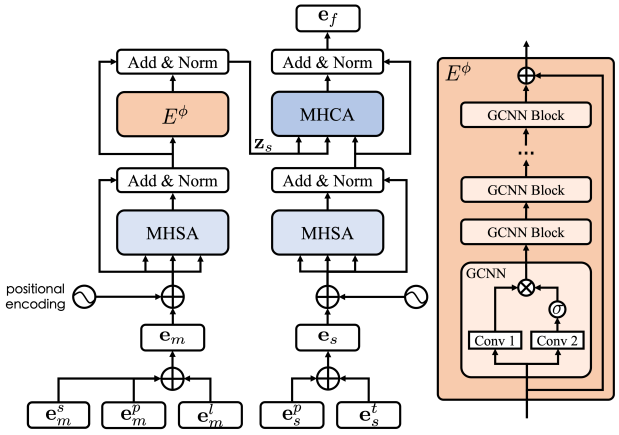


Fig. 3: An illustration of the proposed singing technique local score module (STLS).

we apply Transfer Learning [50] by training our classifier on the larger COSIAN dataset [25] and fine-tuning it on our smaller Opencpop dataset [51] to address limited labeled data. Following Yamamoto et al.'s methodology [25], we used Gaudio Studio[2] to extract singing voice from accompaniment in the COSIAN dataset. For our classification task, we used two softmax functions to handle separate single classification problems, generating distinct labels for pitch and timbre singing techniques.

C. Singing Technique Local Score Module (STLS)

The proposal of the Singing Technique Local Score (STLS) module is motivated by the findings of Lee et al. as documented in their study [20]. They demonstrated that the various components of musical expression in singing can be deduced from a given input text and pitch sequence, implying that the music score inherently encapsulates the elements of singing expression. In the context of our current endeavor, this signifies that the music score inherently embodies characteristics germane to singing technique. To effectively capture the relationship between the singing expression features derived from the music score and the singing technique features, we make modifications to the attention mechanism and introduce the STLS module as our proposed approach. The architecture of STLS module is shown in Fig. 3. As mentioned in III-A, the STLS module accepts five embedding sequences ($\mathbf{e}_m^p$, $\mathbf{e}_m^l$, $\mathbf{e}_m^s$, $\mathbf{e}_s^p$, $\mathbf{e}_s^t$). Primarily, we obtain the music score embedding sequence $\mathbf{e}_m$ by adding $\mathbf{e}_m^p$, $\mathbf{e}_m^l$, $\mathbf{e}_m^s$ and the singing technique embedding sequence $\mathbf{e}_s$ by adding $\mathbf{e}_s^p$, $\mathbf{e}_s^t$. Subsequently, the STLS module employs multi-head self-attention (MHSA) in tandem with residual shortcuts to distill global relationships within the features pertaining to the music score and singing techniques, respectively. Consequently, we introduce a style encoder E^ϕ, which is comprised of ten stacked gated convolutional (GCNN) layers, informed by the methodology proposed in [52], and enriched with skip connections. This architecture is purposed to extract singing expression features denoted as $\mathbf{z}_s$ from the music score embedding $\mathbf{e}_m$. Subsequent to this,

[2]https://studio.gaudiolab.io/gsep

we employ multi-head cross-attention (MHCA) between $\mathbf{z}_s$ and $\mathbf{e}_s$, wherein $\mathbf{e}_s$ serves as the reference, offering queries, while $\mathbf{e}_s$ is regarded as the source, yielding keys and values. Specifically, MHCA facilitates the capture of interdependencies and interactions existing between the singing technique features and the singing expression features, derived from the music score. The STLS module is formulated as shown in (4):

$$\begin{aligned}\mathbf{e}'_m &= \mathrm{LN}(\mathrm{MHSA}(\mathbf{e}_m) + \mathbf{e}_m), \mathbf{e}'_s = \mathrm{LN}(\mathrm{MHSA}(\mathbf{e}_s) + \mathbf{e}_s) \\ \mathbf{z}_s &= \mathrm{LN}(E^{\phi}(\mathbf{e}'_m) + \mathbf{e}'_m) \\ \mathbf{e}_f &= \mathrm{LN}(\mathrm{MHCA}(\mathbf{z}_s, \mathbf{e}'_s) + \mathbf{e}'_s),\end{aligned} \tag{4}$$

where the input order of the $\mathrm{MHCA}(\cdot,\cdot)$ operation is source (K and V) then reference (Q); LN represents layer normalization.

D. Objective Function

To address class imbalance in our Opencpop dataset, analogous to Yamamoto et al.'s findings in the COSIAN dataset, we adapted their methodology by employing a weighted Cross Entropy Loss function with a smoothing factor [41], enhancing model stability. Our objective in this study for singing technique classification is to minimize the weighted cross entropy loss function $\mathcal{L}_{\mathrm{WCE}}$ between the ground truth word-level singing technique labels ($\mathbf{w}$) and the predicted labels ($\hat{\mathbf{w}}$), as described below.

$$\begin{aligned}\mathcal{L}_{\mathrm{WCE}}(\hat{\mathbf{w}}, \mathbf{w}) &= -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} w_c \cdot \mathbf{w}^c \log \hat{\mathbf{w}}^c \\ w_c &= \frac{1}{n_c^{\alpha}},\end{aligned} \tag{5}$$

where n_c denotes the count of training samples within class c, and α represents the smoothing factor that governs the level of smoothing applied to the loss weights.

The loss function of the acoustic model backbone, denoted as $\mathcal{L}_{\mathrm{SVS}}$, closely aligns with the loss function employed in DiffSinger as described in Liu et al. [30], which is founded on the principles of the stable diffusion model. The diffusion process is composed of an equally weighted sequence of denoisers $\epsilon_\theta(\mathbf{M}_t, t, \mathbf{e}_c)$, $t = 1, \cdots, T$, which are trained to predict ϵ added in the diffusion process. The training objective for the singing voice synthesizer can be concisely summarized as follows.

$$\mathcal{L}_{\mathrm{SVS}} = \mathbb{E}_{\mathbf{M},\mathbf{e}_c,\epsilon\sim\mathcal{N}(\mathbf{0},\mathbf{I}),t}\Big[\|\epsilon - \epsilon_\theta(\mathbf{M}_t, t, \mathbf{e}_c)\|_2^2\Big], \tag{6}$$

where $\mathbf{M}_t$ is the noisy mel-spectrogram, t is the time step, $\mathbf{e}_c$ is the music condition sequence.

To achieve the singing technique recommendation, the sequence-to-sequence (S2S) loss $\mathcal{L}_{\mathrm{S2S}}$ is required to train the singing technique decoder D^s. The S2S loss is calculated based on the difference between the ground truth sequence $\mathbf{s} = (s_1, s_2, \cdots, s_T)$ and predicted sequence $\hat{\mathbf{s}} = (\hat{s}_1, \hat{s}_2, ..., \hat{s}_T)$, where T is the length of the sequence. The S2S loss can be formulated as below.

$$\mathcal{L}_{\mathrm{S2S}}(\hat{\mathbf{s}}, \mathbf{s}) = -\frac{1}{T}\sum_{t=1}^{T}\sum_{i=1}^{|V|} \mathbf{s}_{t,i} \log \hat{\mathbf{s}}_{t,i} \tag{7}$$

Algorithm 1 Constructing the Database with Singing Techniques Annotations (Training of STC).

Initialize: θ_{F^s}, θ_{h^o}, θ_{h^c}; $\mathcal{M}$; N_1, N_2, N_3 are number of epochs.

$\mathbf{s}^p_{10}, \mathbf{s}^t_{10} \leftarrow \mathbf{M}_{10}$
for all $i \leftarrow 1$ to N_1 **do**
  repeat
    Sample batch $(\mathbf{M}^c, \mathbf{s}^c)$
    $\mathbf{f}^c \leftarrow F^s(\mathbf{M}^c)$, $\hat{\mathbf{s}^c} \leftarrow h^c(\mathbf{f}^c)$
    Compute $\mathcal{L}_{\mathrm{WCE}}$ based on (5)
    $\theta_{F^s}, \theta_{h^c} \xleftarrow{+} -\nabla_{\theta_{F^s},\theta_{h^c}}(\mathcal{L}_{\mathrm{WCE}})$
  until Batch over
end for
for all $i \leftarrow 1$ to N_3 **do**
  repeat
    Sample batch $(\mathbf{M}_{10}, \mathbf{s}^p_{10}, \mathbf{s}^t_{10})$
    $\mathbf{f}^o \leftarrow F^s(\mathbf{M}_{10})$, $\{\hat{\mathbf{s}^p_{10}}, \hat{\mathbf{s}^t_{10}}\} \leftarrow h^o(\mathbf{f}^o)$
    Compute $\mathcal{L}_{\mathrm{WCE}}$ based on (5)
    if $i \leq N_2$ **then**
      $\theta_{h^o} \xleftarrow{+} -\nabla_{\theta_{h^o}}(\mathcal{L}_{\mathrm{WCE}})$
    else
      $\theta_{F^s}, \theta_{h^o} \xleftarrow{+} -\nabla_{\theta_{F^s},\theta_{h^o}}(\mathcal{L}_{\mathrm{WCE}})$
    end if
  until Batch over
end for
$\mathbf{s}^p_{90}, \mathbf{s}^t_{90} = h^o F^s(\mathbf{M}_{90})$
$\tilde{\mathcal{M}} = \mathcal{M} \cup \{\mathbf{s}^p_{10}, \mathbf{s}^t_{10}\} \cup \{\mathbf{s}^p_{90}, \mathbf{s}^t_{90}\}$

Output: $\tilde{\mathcal{M}}$

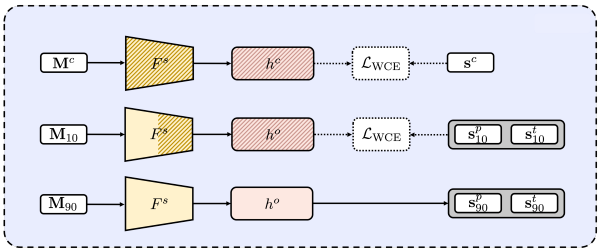
E. Training of SinTechSVS

SinTechSVS follows a multi-step training regimen, depicted in Fig. 4. Initially, we construct the SVS dataset with singing technique annotations by training the singing technique classifier. This involves: (1) manually annotating 10% of the SVS dataset $\mathcal{M}10$ to acquire singing technique labels $\mathbf{s}^p_{10}, \mathbf{s}^t_{10}$; (2) training the classifier with the extensive singing technique dataset $\mathcal{M}^c$; (3) freezing feature extractor F^s and training the header (h^o) of the classifier with $\mathcal{M}_{10}$; (4) unfreezing F^s and training it with $\mathcal{M}_{10}$. Subsequently, the trained classifier infers singing technique labels $\mathbf{s}^p_{90}, \mathbf{s}^t_{90}$ for the remaining 90% of the SVS dataset $\mathcal{M}_{90}$, yielding the annotated SVS dataset $\tilde{\mathcal{M}}$. The algorithm for this process is detailed in Algorithm 1.

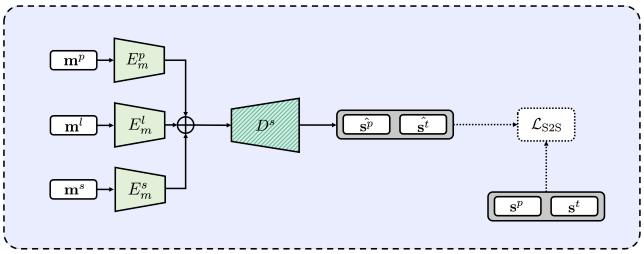
The second step is to train the SVS. During this stage, the encoders, the acoustic model, and the STLS module are optimized using the $\mathcal{L}_{\mathrm{SVS}}$ loss. In the final step, while training the STR, the encoders remain fixed, and only the singing technique decoder D^s is trained. This structured approach ensures the effective training of SinTechSVS, enabling it to generate high-quality singing audio with associated singing techniques.

F. Inference of SinTechSVS

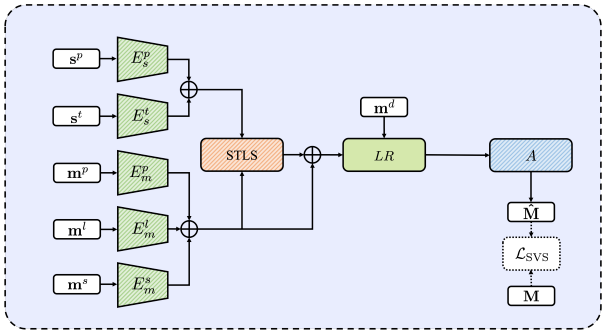
With our meticulously trained models, we are able to synthesize a highly nuanced expressional singing voice through the use of appropriate singing techniques. We offer two distinct inference modes.



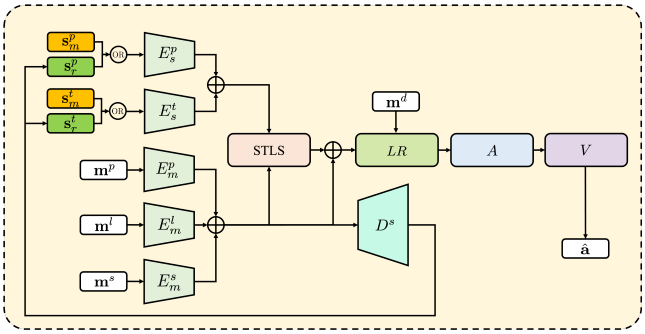
(a) The training of STA. The top line represents training the singing technique classifier using the singing technique dataset ($\mathbf{M}^c$ and $\mathbf{s}^c$). The middle line depicts fine-tuning the classifier with manually annotated SVS dataset ($\mathbf{M}_{10}$, $\mathbf{s}^p_{10}$, and $\mathbf{s}^t_{10}$). The bottom line conducts inference on the remaining SVS dataset to obtain a fully annotated SVS dataset ($\mathbf{M}_{90}$, $\mathbf{s}^p_{90}$, and $\mathbf{s}^t_{90}$).



(b) The training of STR. the training of STR utilizes the pre-trained music score encoder obtained from the previous step (c) and trained the singing technique decoder. $\hat{\mathbf{s}^p}$ and $\hat{\mathbf{s}^t}$ represent the predicted singing technique sequence, whereas $\mathbf{s}^p$ and $\mathbf{s}^t$ represent the annotated singing technique sequence, which obtained from the previous step (a).



(c) The training of SVS. During this step, all modules including the music score encoders, STLS module, and acoustic model undergo training. We utilized annotated SVS dataset, including the singing technique sequence $\mathbf{s}^p$ and $\mathbf{s}^t$ obtained from step (a), the music score sequence $\mathbf{m}^p$, $\mathbf{m}^l$, $\mathbf{m}^s$, $\mathbf{m}^d$, and the mel-spectrogram $\mathbf{M}$.



(d) During the inference phase, singing audios with specific techniques can be synthesized in two ways: (1) manual technique control and (2) technique recommendations based on music scores. $\mathbf{s}^p_m$ and $\mathbf{s}^t_m$ denotes the singing technique sequence manually created, whereas $\mathbf{s}^p_r$ and $\mathbf{s}^t_r$ denotes the singing technique sequence predicted by STR.

Fig. 4: The training process of SinTechSVS consists of three steps, with each step laying the foundation for the next. Modules depicted with full shadows remain unfixed during the training step, while those with half shadows are first fixed and then unfixed during training. The symbol 'OR' represents the logical operation of either given two inputs, while '+' signifies addition. (Top-left): Training of STA; (Bottom-left): Training of STC; (Top-right): Training of STR; (Bottom-left): Training of STA; (Bottom-right): Inference of SinTechSVS.

Manual Singing Technique Control: In this mode, users have the ability to manually control the input singing technique sequences within SinTechSVS. This control permits users to interpret songs and evoke desired expressions by specifying their own singing technique labels. The manual singing technique sequences are represented as $\mathbf{s}^p_m, \mathbf{s}^t_m$.

Predictive Singing Technique Control: Inspired by MsEmoTTS [53], which controlled emotion in speech synthesis through text analysis, we apply a similar method to singing voice synthesis. Using a trained Singing Technique Recommender (STR), we predict singing technique sequences from music scores. In predictive mode, the STR recommends techniques based on input music scores $\mathbf{m}^p, \mathbf{m}^l, \mathbf{m}^s$, resulting in sequences $\mathbf{s}^p_r, \mathbf{s}^t_r$. These sequences are then fed into SinTechSVS for cohesive and expressive singing voice synthesis. This method avoids potential mistakes that can happen with manual skills, giving a strong way to control singing voice synthesis.

IV. Experimental Setups

A. Database Construction

In the previous section, we provided an algorithm 1 to outline the entire flow of our dataset construction process. This section explains our implementation of that algorithm in full detail.

We train our model using the Opencpop dataset [51], comprising 3756 singing utterances from 100 Mandarin songs. Each utterance includes six sequences: word-level and phoneme-level lyrics, phoneme-level pitch, word-level and phoneme-level durations, and slur sequences. We select this dataset for its focus on commercial pop music and its proven effectiveness in training SVS systems, as shown by the success of DiffSinger [30].

We annotated singing technique labels on the Opencpop dataset following the method outlined in Sections III-A and III-B, involving three semi-professional musicians: one with 14 years of piano experience and 10 years in choir and conducting, another with 10 years in POP music singing, and the third, a vocalist of jazz band, with three years of jazz singing. Using mel-spectrogram representations from [25] as benchmarks, we assigned pitch and timbral technique labels

based on the closest match. The labeled dataset distribution is shown in Table III.

To enhance the robustness of the dataset, we apply the following data processing steps: (1) Given that the COSIAN dataset comprises stereo audio files and the Opencpop dataset contains mono audio files, we standardize all files to stereo to avoid potential information loss; (2) Remove silence by clipping sections with waveform values below a specified threshold to eliminate unwanted noise and enhance segment clarity; (3) Normalize audio segments using pydub[3] for consistency and stable training; (4) Convert audio segments into mel-spectrograms with 64 mel bands, a window size of 1024, and a sample rate of 44.1kHz, scaled to a decibel range with a minimum cut-off of -80db; (5) We divided training mel-spectrograms into short (0-100), medium (101-200), and long (201-300) groups based on duration, padding each to its maximum length within the respective bin to minimize zero-padding, facilitating efficient batch training and exposure to multi-sized inputs during training [49]. To clarify, during SinTechSVS acoustic model training, we resample all Opencpop audio samples to 24kHz to meet HiFiGAN requirements and expedite training, despite the singing technique classifier being trained on 44.1kHz mel-spectrograms.

The subsequent paragraphs detail the characteristics of our manually annotated Opencpop dataset.

Fig. 5 illustrates the joint distribution of phonemes and pitch within the manually annotated section of the Opencpop dataset. Our random sampling covers a wide pitch and phoneme domain, highlighting key pitch-phoneme pairings. Furthermore, comparing Fig. 6 with Figure 1 in [25] our observation underscores the language independence of the chosen singing techniques, as evidenced by the resemblance between the mel-spectrograms of the COSIAN and Opencpop datasets. This aligns with our successful classifier transfer from a Japanese to a Chinese singing dataset.

To train our Singing Technique Classifier, we utilized a subset of the Opencpop dataset and the COSIAN dataset, allocating 15% for validation and 85% for training. To address the imbalance in technique distribution, particularly in songs, we applied a smooth weighted Cross Entropy Loss method discussed in Section III-D.

TABLE III: Distribution of manually annotated portion of Opencpop dataset. The singing techniques "whisper" and "hiccup" are removed due to the small amount of labels.

Singing Technique	Train	Test	Total
Falsetto	628	111	739
Breathy	765	135	900
Vocal Fry	63	11	74
Belting	1273	225	1498
Regular	1289	228	1517
Whisper	-	-	12
Scooping	864	152	1016
Drop	161	28	189
Bend	108	19	127
Melisma	68	12	80
Straight	2819	497	3316
Hiccup	-	-	4
Total	8038	1418	9456

[3]http://pydub.com/

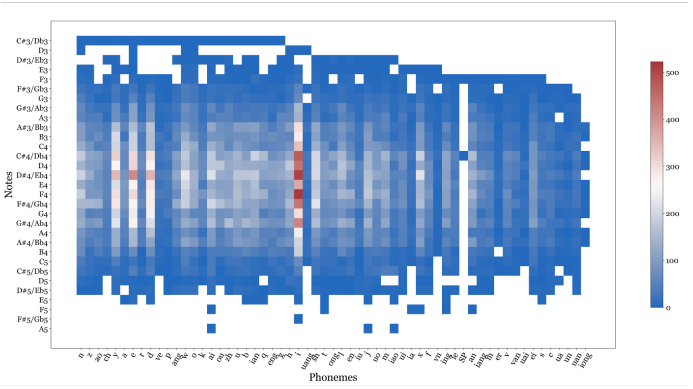


Fig. 5: The joint distribution of the phoneme and the pitch of the singing techniques.

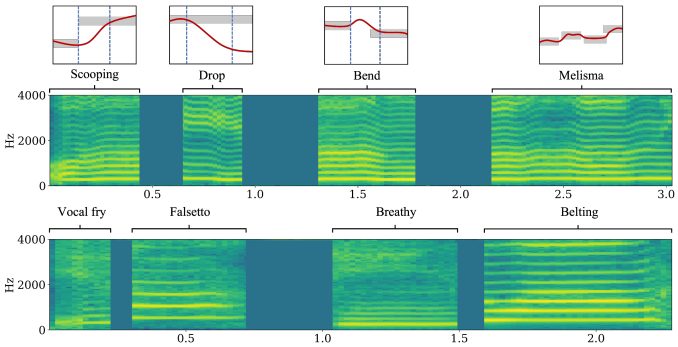


Fig. 6: The mel-spectrograms of the timbral and pitch singing techniques with a sketch of pitch contour sampled from the manually annotated Opencpop dataset.

B. Implementation Details

In the course of our experiments, SinTechSVS and all other SVS models in the ablation studies are trained using a single RTX-A5000 GPU equipped with 23GB of memory. Here, we provide a comprehensive breakdown of the training process for the SVS models. The initial phase of our experimentation involved training the classifier, which was accomplished by following the procedures outlined in Algorithm 1. The COSIAN dataset served as the training dataset for this phase. The following hyperparameters were employed: (1) Adam optimizer with learning rate = 10^{-3} and weight decay = 0.03, (2) batch size = 32; (3) epochs = 100. Subsequently, transfer learning was applied to fine-tune the pre-trained classifier using the Opencpop dataset. During this fine-tuning process, we adjusted the hyperparameters as follows: (1) Adam optimizer with learning rate = 0.0001 and weight decay = 0.05, (2) batch size = 32; (3) epochs = 73. In both stages of classifier training, we implemented a learning rate reduction strategy. Specifically, we reduced the learning rate by half whenever there were three consecutive epochs with no observed increase in validation accuracy.

The subsequent phase of our experiments involved training the singing technique synthesizer and recommender, in accordance with the procedures outlined in Section III-E. This training process spanned a duration of 12 hours and utilized the following hyperparameters: (1) Adam optimizer with learning rate = 10^{-3}; (2) batch size = 48; (3) epochs = 160K.

C. Evaluation Methods

In this section, we outline the various evaluation methods used to assess the synthesis capabilities and singing technique controllability of SinTechSVS[4].

Synthesis Capability Evaluation: For objective evaluation, we employed Mel Cepstral Distortion (MCD) [54] and Root Mean-Squared Error of Fundamental Frequency (F0-RMSE) [55]. For subjective evaluation, we conducted a Mean Opinion Score (MOS) survey with formal singing training and strong knowledge of singing techniques [21], [16], [17] to evaluate the naturalness and sound quality of singing voices synthesized by various SVS models. The survey used a 5-point scale (with 5 being the highest) to rate the overall pleasantness of the synthesized outputs as evaluated by 20 participants. Participants rated each audio sample for naturalness, pronunciation accuracy, and overall quality. The MOS survey results will be further discussed in Sections V-A and V-B.

Singing Technique Controllability Evaluation: We evaluated SinTechSVS controllability using two metrics: Style Reclassification Accuracy ($\mathrm{SR}_{\mathrm{acc}}$) and Style Match Rate (SMR), inspired by style conditioned synthesis [56]. $\mathrm{SR}_{\mathrm{acc}}$ measures accuracy in reclassifying synthesized audio clips' singing techniques, whereas SMR assesses alignment between input labels and listener perceptions of the technique class.

Style Reclassification Accuracy ($\mathrm{SR}_{\mathrm{acc}}$) can be formulated as in (8):

$$\mathrm{SR}_{\mathrm{acc}}(\mathbf{x}) = \frac{1}{M}\sum_{i=1}^{M}\frac{1}{N_m}\sum_{i=1}^{N_m}\mathbb{1}\{C(x_{i,m}) = m\}, \tag{8}$$

where $\mathbf{x}$ is the synthesized audio clips, C represents the singing technique classifier, N is the number of audio samples, M is the number of singing technique classes.

We conduct Style Match Rate (SMR) in two tasks: (1) SMR for singing technique identification (SMR-I) and (2) SMR for singing technique matching (SMR-M). Both use a generalized accuracy formula as described in (9):

$$\mathrm{SMR}(\mathbf{r}, \mathbf{y}) = \frac{\sum_{i=1}^{N}\mathbb{1}\{r_i = y_i\}}{N}, \tag{9}$$

where $\mathbf{r}$ is the listener's response based on their assessment of the synthesized audio clips with specific singing techniques at a word or sentence level, $\mathbf{y}$ is the correct answer for each survey question, and N is the number of samples.

The MOS survey consisted of four distinct segments, where participants were tasked with identifying singing techniques or matching singing techniques to the corresponding audio samples:

- Participants identify pitch singing techniques in highlighted words in audio samples.
- Participants match audio samples with their correct pitch singing techniques by comparing two samples with different techniques.
- Participants identify timbral singing techniques in highlighted words in audio samples.
- Participants match audio samples with their correct timbral singing techniques by comparing two samples with different techniques.

[4]This study has been approved by the Department Ethics Review Committee (DERC) at the National University of Singapore under soc-23-32

TABLE IV: The objective comparison of unconditional singing voice synthesis between SinTechSVS, various SOTA SVS, and Ground Truth audio samples from Opencpop (GT). Pron. Acc. refers to Pronunciation Accuracy

Model	MCD	F0-RSME	Naturalness	Quality	Pron. Acc.
GAN-Singer	4.95	0.0252	2.97 ± 0.15	3.48 ± 0.19	3.57 ± 0.17
CpopSing	4.93	0.0256	3.02 ± 0.21	3.26 ± 0.09	3.79 ± 0.11
DiffSinger	4.86	0.0233	3.27 ± 0.16	3.52 ± 0.12	**4.13 ± 0.16**
$\mathrm{SinTechSVS}_{\mathrm{STR}}$	-	-	**3.21 ± 0.15**	**3.54 ± 0.21**	**3.82 ± 0.20**
SinTechSVS	**4.71**	**0.0228**	**3.29 ± 0.20**	**3.72 ± 0.15**	**4.02 ± 0.10**
GT	-	-	4.09 ± 0.18	4.25 ± 0.18	4.38 ± 0.14

D. Compared Methods

We compare SinTechSVS against state-of-the-art SVS systems like CpopSing [51], GAN-Singer [57], and DiffSinger [30]. For a fair comparison, we employed the same vocoder (HiFi-GAN) across all systems. Given the absence of available singing technique-controllable systems for direct comparison, we evaluated our system's controllability with evaluation metrics described in IV-C, using ground truth audio samples as performance ceilings.

V. Experimental Results

A. Results of Unconditional Singing Voice Synthesis

Table IV showcases our SVS system's performance, presenting scores for Naturalness, Quality, Pronunciation Accuracy, and overall MOS, each with a 95% confidence interval. We conducted a comprehensive comparison with Ground Truth samples, GAN-Singer [58], CpopSing [51], and DiffSinger [30].

Ground Truth samples unsurprisingly outperformed across all subjective metrics. However, our models demonstrated comparable performance to DiffSinger and outperformed in terms of MCD and F0-RMSE. These results underscore the effectiveness of conditioning our system on implicit expression features, as suggested in [1], to enhance naturalness and robustness.

Additionally, we also compared our models with $\mathrm{SinTechSVS}_{\mathrm{STR}}$, where singing technique sequences are predicted by the STR. Results indicate that $\mathrm{SinTechSVS}_{\mathrm{STR}}$ performs slightly worse than both DiffSinger and SinTechSVS (slightly better in Quality compared to DiffSinger), but better than other baseline models as demonstrated by the baseline models.

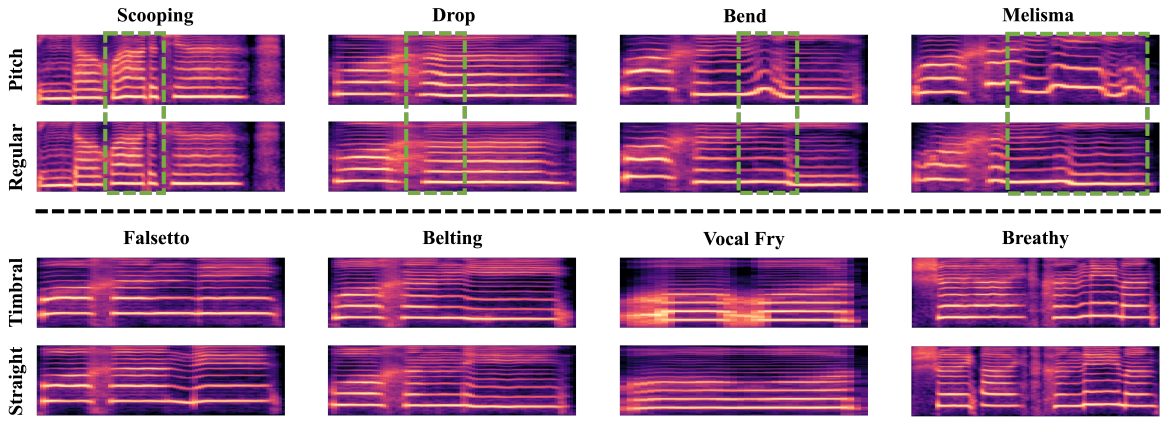


Fig. 7: Comparing mel-spectrograms of audio samples generated by SinTechSVS with distinct pitch and timbral singing techniques against the corresponding audios generated by SinTechSVS with no techniques applied. 'Straight' represents mel-spectrograms without pitch techniques, while 'Regular' showcases mel-spectrograms without timbre techniques. The green bounding boxes highlight regions demonstrating pitch singing technique control.

TABLE V: Results of Conditional Singing Voice Synthesis. $\mathrm{ST_{an}}$ refers to singing technique annotated labels.

Singing Technique	$\mathrm{SR_{acc}}$		SMR-I		SMR-M	
	$\mathrm{ST_{an}}$	SinTechSVS	$\mathrm{ST_{an}}$	SinTechSVS	$\mathrm{ST_{an}}$	SinTechSVS
Falsetto	90%	82%	68%	56%	85%	78%
Breathy	76%	63%	72%	70%	88%	80%
Vocal Fry	79%	71%	64%	62%	84%	74%
Belting	80%	74%	76%	84%	90%	94%
Scooping	85%	82%	84%	76%	92%	88%
Drop	77%	71%	82%	78%	86%	82%
Bend	52%	66%	71%	79%	82%	85%
Melisma	39%	51%	80%	66%	74%	82%
Average	72%	70%	75%	71%	85%	83%

B. Results of Singing Voice Synthesis Conditioned on Singing Techniques by Manual Control

Table V displays results for Style-Reclassification Accuracy ($\mathrm{SR_{acc}}$) and Style Match Rate (SMR) obtained from SinTechSVS generated audio samples compared to singing technique annotated labels from the Opencpop dataset ($\mathrm{ST_{an}}$).

Our model's $\mathrm{SR_{acc}}$ values closely align with those of $\mathrm{ST_{an}}$, indicating the singing technique classifier can effectively discern between the various singing techniques generated by SinTechSVS.

In the SMR results, as anticipated, participants excelled in matching tasks, known for their lower complexity. Our top-performing model demonstrates precise control over singing techniques, as evidenced by SMR values closely resembling those of $\mathrm{ST_{an}}$. This highlights SinTechSVS's proficiency in generating highly accurate and distinctive singing techniques.

In Fig. 7, we compare mel-spectrograms from audio samples with distinct singing techniques against those generated by SinTechSVS without specific techniques. This visual evidence highlights SinTechSVS's effectiveness in controlling singing techniques.

TABLE VI: Comparison of MOS results of SinTechSVS conditioned on singing techniques from various sources. Pron. Acc. refers to pronunciation accuracy.

Model	Naturalness	Quality	Pron. Acc.
$\mathrm{SinTechSVS_{Rand}}$	2.39 ± 0.19	2.02 ± 0.20	3.56 ± 0.19
$\mathrm{SinTechSVS_{Norm}}$	3.00 ± 0.16	2.85 ± 0.18	3.66 ± 0.23
$\mathrm{SinTechSVS_{STR}}$	$\mathbf{3.21 \pm 0.15}$	$\mathbf{3.54 \pm 0.21}$	$\mathbf{3.82 \pm 0.20}$
$\mathrm{SinTechSVS_{GT}}$	3.29 ± 0.20	3.72 ± 0.20	4.02 ± 0.22

C. Results of Singing Voice Synthesis Conditioned on Singing Techniques by Singing Technique Recommendation

To demonstrate the effectiveness of the singing technique recommender, we compare the synthesis results of SinTechSVS with different inputs:

- $\mathrm{SinTechSVS_{Rand}}$: the singing technique sequences input is randomly generated $\{\mathbf{s}^p_{rd}, \mathbf{s}^t_{rd}\}$.
- $\mathrm{SinTechSVS_{Norm}}$: the singing technique sequences input with no singing techniques $\{\mathbf{s}^p_n, \mathbf{s}^t_n\}$. This is achieved by assigning the pitch singing technique sequence to be all 'Straight' and assigning the timbral singing technique sequence to be all 'Regular'.
- $\mathrm{SinTechSVS_{STR}}$: the singing technique sequences input is generated from the corresponding music score sequence $\{\mathbf{m}^p, \mathbf{m}^l, \mathbf{m}^s\}$ by using STR $\{\mathbf{s}^p_r, \mathbf{s}^t_r\}$.
- $\mathrm{SinTechSVS_{GT}}$: the singing technique sequences input is the original annotated singing technique sequence $\{\mathbf{s}^p, \mathbf{s}^t\}$. $\mathrm{SinTechSVS_{GT}}$ is actually same as SinTechSVS mentioned in Table IV.

Table VI highlights the system's performance across these scenarios. SinTechSVSSTR lags slightly behind $\mathrm{SinTechSVS_{GT}}$ but outperforms $\mathrm{SinTechSVS_{Rand}}$ significantly, emphasizing the value of the singing technique recommender. $\mathrm{SinTechSVS_{Norm}}$ achieves superior scores to $\mathrm{SinTechSVS_{Rand}}$ but falls short compared to $\mathrm{SinTechSVS_{GT}}$ and $\mathrm{SinTechSVS_{STR}}$, underscoring the importance of fine-grained control over singing techniques in SVS systems.

TABLE VII: The evaluation results of Conditional Singing Voice Synthesis across ablated models. T-Average represents the average value of the timbral singing techniques of a model across a metric, while P-Average represents the average value of the pitch singing techniques of a model across a metric.

Singing Technique	SR_{acc}				SMR-I				SMR-M			
	SinTechSVS	w/o-STLS	w/o-STLS$_p$	w/o-STLS$_t$	SinTechSVS	w/o-STLS	w/o-STLS$_p$	w/o-STLS$_t$	SinTechSVS	w/o-STLS	w/o-STLS$_p$	w/o-STLS$_t$
Falsetto	**82%**	74%	78%	80%	56%	62%	**64%**	53%	78%	73%	**85%**	71%
Breathy	63%	58%	65%	**67%**	**70%**	65%	66%	59%	80%	75%	77%	**82%**
Vocal Fry	71%	67%	**73%**	65%	62%	57%	**71%**	67%	**74%**	66%	70%	65%
Belting	**74%**	68%	62%	59%	**84%**	81%	78%	69%	**94%**	80%	88%	91%
T-Average	**73%**	67%	70%	68%	68%	66%	**70%**	62%	**82%**	74%	80%	78%
Scooping	**82%**	76%	67%	80%	76%	35%	46%	**79%**	**88%**	62%	56%	87%
Drop	**71%**	66%	68%	70%	**78%**	57%	42%	77%	82%	67%	74%	**84%**
Bend	66%	**69%**	64%	60%	**79%**	31%	39%	70%	**85%**	58%	70%	79%
Melisma	51%	41%	37%	**56%**	66%	43%	51%	**69%**	**82%**	74%	65%	76%
P-Average	**68%**	63%	59%	67%	**75%**	42%	45%	74%	**85%**	65%	67%	82%
Average	**71%**	65%	65%	68%	**72%**	54%	58%	68%	**84%**	70%	74%	80%

While we demonstrate STR's effectiveness, it still falls short of surpassing all metrics compared to all the other baseline models, as shown in Table IV. This could be attributed to the lack of a high-quality, large dataset for training STR, which severely limits its capabilities. Besides, the current STR is designed to predict singing technique sequences with a limited length, as it was trained solely on the singing data with utterance length. However, recommending singing techniques with varying lengths, such as song-level or utterance-level, is a topic we aim to explore in future research endeavors.

D. Ablations of Singing Technique Classifier Architecture

We compare several different architectures for the ablation study. (STC = Singing Technique Classifier)

- OblongSTC: a CNN that uses oblong convolutional filters proposed by Yamamoto et al. [41].
- StandardSTC: a CNN that uses the architecture and parameters outlined in Table II represents the ultimate choice for our STC implementation.
- ResSTC: a CNN that uses residual blocks similar to ResNet50 [59] for its success in the field of Computer Vision.

We trained two versions of each model: one with weighted loss (smoothing factor of 0.33) and one without weights, to assess the impact on learning from imbalanced data. Additionally, we tested transfer learning by training two more versions of StandardSTC without pre-training on the COSIAN dataset. The architecture of StandardSTC is detailed in Section III-B.

We evaluated each model with the following metrics for both timbral and pitch techniques: Accuracy (Acc.), Top-2 Accuracy, Balanced Accuracy (BAcc.), Macro-F1 score (F1).

Table VIII and Fig. 8 show the evaluation results of classifier models trained on pitch and timbral data. Initially, we tried replicating Yamamoto et al.'s [25] Singing Technique Classifier model but found it unsuitable. We then explored a standard approach using 3×3 and 5×5 convolutional filters. The StandardSTC model with a weighted loss function outperformed the OblongSTC model across all metrics. Despite experimenting with residual blocks to enhance performance (ResSTC model), it didn't surpass the StandardSTC model,

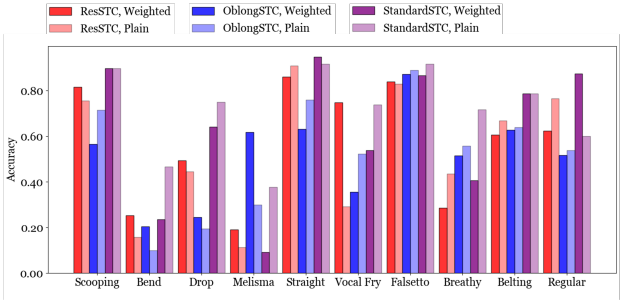


Fig. 8: Comparison of per-class accuracy across different classifiers.

likely due to limited training data. Additionally, pre-trained models showed a minimum 0.1 improvement across all metrics, confirming the benefits of transfer learning.

E. Ablations of Singing Technique Local Score Module (STLS)

In this ablation study, we explore the impact of the Singing Technique Local Score (STLS) module on SinTechSVS performance across four configurations:

- SinTechSVS: This represents our proposed SinTechSVS system with all components intact.
- -w/o-STLS: In this setup, we remove the STLS module from SinTechSVS, resulting in the condition where $\mathbf{e}_c = LR(\mathbf{e}_m + \mathbf{e}_s, \mathbf{m}^d)$, where $\mathbf{e}_m = \mathbf{e}_m^p + \mathbf{e}_m^l + \mathbf{e}_m^s$, and

TABLE VIII: Results of singing technique classification with different model configurations. Models with (*) were not pre-trained on the COSIAN dataset

Model	Acc.	Top-2	BAcc.	F1
OblongSTC	0.665	0.878	0.521	0.437
OblongSTC, weighted	0.596	0.849	0.515	0.411
ResSTC	0.752	0.912	0.517	0.476
ResSTC, weighted	0.700	0.884	0.571	0.469
StandardSTC	**0.820**	0.948	0.628	0.551
StandardSTC, weighted	0.808	**0.952**	**0.723**	**0.588**
StandardSTC*	0.600	0.724	0.557	0.427
StandardSTC*, weighted	0.692	0.811	0.622	0.483

TABLE IX: The evaluation results of unconditional singing voice synthesis across ablated models. Pron. Acc. refers to Pronunciation Accuracy

Model	MCD	F0-RSME	Naturalness	Quality	Pron. Acc.
SinTechSVS	**4.71**	0.0228	3.29 ± 0.20	**3.72 ± 0.15**	4.02 ± 0.10
w/o-STLS	4.82	**0.0226**	3.17 ± 0.12	3.66 ± 0.18	3.90 ± 0.15
w/o-STLS$_p$	4.73	0.0233	**3.36 ± 0.19**	3.59 ± 0.13	3.86 ± 0.17
w/o-STLS$_t$	4.77	0.0241	3.25 ± 0.13	3.70 ± 0.14	**4.07 ± 0.16**

$\mathbf{e}_s = \mathbf{e}_s^p + \mathbf{e}_s^t$.

- -w/o-STLS$_p$: Here, we keep the pitch singing technique sequence from the STLS module operation, leading to $\mathbf{e}_f = \mathrm{STLS}(\mathbf{e}_m, \mathbf{e}_s^t) + \mathbf{e}_s^p$.
- -w/o-STLS$_t$: In this configuration, we keep the timbral singing technique sequence from the STLS module operation, leading to $\mathbf{e}_f = \mathrm{STLS}(\mathbf{e}_m, \mathbf{e}_s^p) + \mathbf{e}_s^t$.

One thing noteworthy is that all comparison models maintain access to singing technique information $\mathbf{e}_s$ and music score information $\mathbf{e}_m$, even in the absence of utilizing the STLS module. Results in Table IX show comparable synthesis capabilities across all configurations, suggesting a minimal influence of STLS on general synthesis.

Evaluation then focuses on STLS's effect on controllability. Table VII shows that SinTechSVS with STLS consistently achieves the highest scores in pitch and timbral tasks, indicating significant improvement in controllability during synthesis. Notably, The STLS module is especially effective for tasks related to pitch. This makes sense because musical scores directly contain pitch information, whereas timbral details are more inherent and subtle.

VI. Conclusion

In this paper, we present SinTechSVS, an innovative Singing Voice Synthesis (SVS) system that offers precise control over singing techniques. We begin by introducing a data-efficient method for singing technique annotation. Specifically, we manually annotate 10% of the Opencpop dataset and construct a CNN-based singing technique classifier with a Temporal Pyramid Pooling (TPP) layer to infer labels for the rest of the Opencpop dataset. Subsequently, we extend the SVS system to be controllable in terms of singing techniques by incorporating style encoders and proposing an attention-based singing technique local score (STLS) module, which has been shown to enhance the system's ability to accurately control singing techniques during synthesis. Furthermore, we develop a Seq2Seq singing technique recommender that can recommend appropriate singing techniques based on the music score. Our series of experiments demonstrate that SinTechSVS exhibits satisfactory performance in terms of both general synthesis capability and the controllability of singing techniques while synthesizing singing voices.

References

[1] Y.-P. Cho, F.-R. Yang, Y.-C. Chang, C.-T. Cheng, X.-H. Wang, and Y.-W. Liu, "A survey on recent deep learning-driven singing voice synthesis systems," in *2021 IEEE International Conference on Artificial Intelligence and Virtual Reality (AIVR)*. IEEE, 2021, pp. 319–323, doi: 10.1109/AIVR52153.2021.00067.

[2] R. Kobayashi, "Sound clustering synthesis using spectral data," in *ICMC*, 2003. [Online]. Available: https://nagasm.org/ASL/icmc2003/closed/CR1052.PDF

[3] M. Puckette, "Low-dimensional parameter mapping using spectral envelopes." in *ICMC*, 2004. [Online]. Available: https://msp.ucsd.edu/Publications/icmc04.pdf

[4] R. Hoskinson, "Manipulation and resynthesis of environmental sounds with natural wavelet grains," Ph.D. dissertation, University of British Columbia, 2002, doi: 10.14288/1.0051535.

[5] S. Dubnov, Z. Bar-Joseph, R. El-Yaniv, D. Lischinski, and M. Werman, "Synthesizing sound textures through wavelet tree learning," *IEEE Comput. Graph. Appl.*, vol. 22, no. 4, p. 38–48, jul 2002, doi: 10.1109/MCG.2002.1016697.

[6] P. Xiang, "A new scheme for real-time loop music production based on granular similarity and probability control," in *Proceedings of the International Conference on Digital Audio Effects*, 2002, pp. 89–92. [Online]. Available: https://www.dafx.de/papers/DAFX02_Xiang_music_production.pdf

[7] P. Cano, L. Fabig, F. Gouyon, M. Koppenberger, A. Loscos, and A. Barbosa, "Semi-automatic ambiance genereation," in *Proceedings of the International Conference on Digital Audio Effects, DAFx*, 2004, pp. 319–322, doi:a2988a3ef5375bdf6456eb171259c13d08f464b2.

[8] B. L. Sturm, "Concatenative sound synthesis and intellectual property: An analysis of the legal issues surrounding the synthesis of novel sounds from copyright-protected work," *Journal of New Music Research*, vol. 35, no. 1, pp. 23–33, 2006, doi:https://doi.org/10.1080/09298210600696691.

[9] G. Beller, D. Schwarz, T. Hueber, and X. Rodet, "Hybrid concatenative synthesis on the intersection of music and speech," in *Journées d'informatique musicale*, 2005, pp. 41–45. [Online]. Available: https://hal.science/hal-01161411v1/document

[10] K. Saino, H. Zen, Y. Nankaku, A. Lee, and K. Tokuda, "An HMM-based singing voice synthesis system," in *Proc. Interspeech 2006*, 2006, pp. paper 2077–Thu1BuP.7, doi:10.21437/Interspeech.2006-584.

[11] K. Oura, A. Mase, T. Yamada, S. Muto, Y. Nankaku, and K. Tokuda, "Recent development of the hmm-based singing voice synthesis system—sinsy," in *Seventh ISCA Workshop on Speech Synthesis*, 2010. [Online]. Available: https://www.isca-archive.org/ssw_2010/oura10_ssw.pdf

[12] K. Oura, A. Mase, Y. Nankaku, and K. Tokuda, "Pitch adaptive training for hmm-based singing voice synthesis," in *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2012, pp. 5377–5380, doi:10.1109/ICASSP.2014.6854062.

[13] K. Shirota, K. Nakamura, K. Hashimoto, K. Oura, Y. Nankaku, and K. Tokuda, "Integration of speaker and pitch adaptive training for hmm-based singing voice synthesis," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2014, pp. 2559–2563, doi:10.1109/ICASSP.2012.6289136.

[14] T. Nose, M. Kanemoto, T. Koriyama, and T. Kobayashi, "Hmm-based expressive singing voice synthesis with singing style control and robust pitch modeling," *Computer Speech & Language*, vol. 34, no. 1, pp. 308–322, 2015, doi:https://doi.org/10.1016/j.csl.2015.04.001.

[15] Y. Ren, X. Tan, T. Qin, J. Luan, Z. Zhao, and T.-Y. 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*, 2020, pp. 1979–1989, doi:https://doi.org/10.1145/3394486.3403249.

[16] P. Lu, J. Wu, J. Luan, X. Tan, and L. Zhou, "Xiaoicesing: A high-quality and integrated singing voice synthesis system," *arXiv preprint arXiv:2006.06261*, 2020, doi:https://doi.org/10.48550/arXiv.2006.06261.

[17] C. Wang, C. Zeng, and X. He, "Xiaoicesing 2: A high-fidelity singing voice synthesizer based on generative adversarial network," *arXiv preprint arXiv:2210.14666*, 2022, doi:10.48550/arXiv.2210.14666.

[18] M. Umbert, J. Bonada, M. Goto, T. Nakano, and J. Sundberg, "Expression control in singing voice synthesis: Features, approaches, evaluation, and challenges," *IEEE Signal Processing Magazine*, vol. 32, no. 6, pp. 55–73, 2015, doi:10.1109/MSP.2015.2424572.

[19] K. R. Scherer, J. Sundberg, B. Fantini, S. Trznadel, and F. Eyben, "The expression of emotion in the singing voice: Acoustic patterns in vocal performance," *The Journal of the Acoustical Society of America*, vol. 142, no. 4, pp. 1805–1815, 2017, doi:10.1121/1.5002886.

[20] J. Lee, H.-S. Choi, and K. Lee, "Expressive singing synthesis using local style token and dual-path pitch encoder," *arXiv preprint arXiv:2204.03249*, 2022. [Online]. Available: https://arxiv.org/pdf/2204.03249.pdf

[21] S. Kim, Y. Kim, J. Jun, and I. Kim, "Muse-svs: Multi-singer emotional singing voice synthesizer that controls emotional intensity," *IEEE/ACM*

Transactions on Audio, Speech, and Language Processing, vol. 31, pp. 2751–2764, 2023, doi:10.1109/TASLP.2023.3294712.

[22] Y. Song, W. Song, W. Zhang, Z. Zhang, D. Zeng, Z. Liu, and Y. Yu, "Singing voice synthesis with vibrato modeling and latent energy representation," in *2022 IEEE 24th International Workshop on Multimedia Signal Processing (MMSP)*. IEEE, 2022, pp. 1–6, doi:10.1109/MMSP55362.2022.9948936.

[23] J. Lee, H.-S. Choi, J. Koo, and K. Lee, "Disentangling timbre and singing style with multi-singer singing synthesis system," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 7224–7228, doi:10.1109/ICASSP40776.2020.9054636.

[24] S. Resna and R. Rajan, "Multi-voice singing synthesis from lyrics," *Circuits, Systems, and Signal Processing*, vol. 42, no. 1, pp. 307–321, 2023, doi:10.1007/s00034-022-02122-3.

[25] Y. Yamamoto, J. Nam, and H. Terasawa, "Analysis and detection of singing techniques in repertoires of j-pop solo singers," in *Proceedings of the 23rd International Society for Music Information Retrieval Conference (ISMIR)*, 2022, doi:https://doi.org/10.48550/arXiv.2210.17367.

[26] H. Kenmochi and H. Ohshita, "Vocaloid-commercial singing synthesizer based on sample concatenation," in *Interspeech*, vol. 2007, 2007, pp. 4009–4010. [Online]. Available: https://api.semanticscholar.org/CorpusID:17345450

[27] J. Bonada and X. Serra, "Synthesis of the singing voice by performance sampling and spectral models," *IEEE signal processing magazine*, vol. 24, no. 2, pp. 67–79, 2007, doi:10.1109/MSP.2007.323266.

[28] Y. Hono, K. Hashimoto, K. Oura, Y. Nankaku, and K. Tokuda, "Sinsy: A deep neural network-based singing voice synthesis system," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 2803–2815, 2021, doi:https://doi.org/10.1109/TASLP.2021.3104165.

[29] J. Chen, X. Tan, J. Luan, T. Qin, and T.-Y. Liu, "Hifisinger: Towards high-fidelity neural singing voice synthesis," *arXiv preprint arXiv:2009.01776*, 2020, doi:https://doi.org/10.48550/arXiv.2009.01776.

[30] J. Liu, C. Li, Y. Ren, F. Chen, and Z. Zhao, "Diffsinger: Singing voice synthesis via shallow diffusion mechanism," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 10, 2022, pp. 11 020–11 028, doi:10.1609/aaai.v36i10.21350.

[31] M. Blaauw and J. Bonada, "A neural parametric singing synthesizer modeling timbre and expression from natural songs," *Applied Sciences*, vol. 7, no. 12, 2017, doi:10.3390/app7121313.

[32] R. Liu, X. Wen, C. Lu, L. Song, and J. S. Sung, "Vibrato learning in multi-singer singing voice synthesis," in *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2021, pp. 773–779, doi:10.1109/ASRU51503.2021.9688029.

[33] Y. Wu, S. Li, C. Yu, H. Lu, C. Weng, L. Zhang, and D. Yu, "Synthesising expressiveness in peking opera via duration informed attention network," *arXiv preprint arXiv:1912.12010*, 2019, doi:10.21437/Interspeech.2020-1724.

[34] Y. Ikemiya, K. Itoyama, and H. G. Okuno, "Transferring vocal expression of f0 contour using singing voice synthesizer," in *Modern Advances in Applied Intelligence: 27th International Conference on Industrial Engineering and Other Applications of Applied Intelligent Systems, IEA/AIE 2014, Kaohsiung, Taiwan, June 3-6, 2014, Proceedings, Part II 27*. Springer, 2014, pp. 250–259, doi:https://doi.org/10.1007/978-3-319-07467-$2_2$7.

[35] T. Nakano and M. Goto, "Vocalistener2: A singing synthesis system able to mimic a user's singing in terms of voice timbre changes as well as pitch and dynamics," in *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2011, pp. 453–456, doi:10.1109/ICASSP.2011.5946438.

[36] P. Proutskova, C. Taveras, and Y. N. Hung, "Phonation modes dataset," Mar 2022. [Online]. Available: osf.io/pa3ha

[37] J. Wilkins, P. Seetharaman, A. Wahl, and B. Pardo, "Vocalset: A singing voice dataset," in *International Society for Music Information Retrieval Conference*, 2018, doi:10.5281/zenodo.1193957.

[38] K. L. Kim, J. Lee, S. Kum, C. L. Park, and J. Nam, "Semantic tagging of singing voices in popular music recordings," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 1656–1668, 2020, doi:10.1109/TASLP.2020.2993893.

[39] V. Kalbag and A. Lerch, "Scream detection in heavy metal music," 2022, doi:https://doi.org/10.48550/arXiv.2205.05580.

[40] H. K. Schutte and D. G. Miller, "Belting and pop, nonclassical approaches to the female middle voice: some preliminary considerations," *Journal of Voice*, vol. 7, no. 2, pp. 142–150, 1993, doi:https://doi.org/10.1016/S0892-1997(05)80344-3.

[41] Y. Yamamoto, J. Nam, and H. Terasawa, "Deformable cnn and imbalance-aware feature learning for singing technique classification," *arXiv preprint arXiv:2206.12230*, 2022, doi:10.21437/Interspeech.2022-11137.

[42] Z. Zhang, Y. Zheng, X. Li, and L. Lu, "Wesinger: Data-augmented singing voice synthesis with auxiliary losses," *arXiv preprint arXiv:2203.10750*, 2022, doi:10.48550/arXiv.2203.10750.

[43] Y. Gu, X. Yin, Y. Rao, Y. Wan, B. Tang, Y. Zhang, J. Chen, Y. Wang, and Z. Ma, "Bytesing: A chinese singing voice synthesis system using duration allocated encoder-decoder acoustic models and wavernn vocoders," in *2021 12th International Symposium on Chinese Spoken Language Processing (ISCSLP)*. IEEE, 2021, pp. 1–5, doi:10.1109/ISCSLP49672.2021.9362104.

[44] J. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," in *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, ser. NIPS'15. Cambridge, MA, USA: MIT Press, 2015, p. 577–585, doi:https://doi.org/10.48550/arXiv.1506.07503. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2015/file/1068c6e4c8051cfd4e9ea8072e3189e2-Paper.pdf

[45] B. Zhang, J. Leitner, and S. Thornton, "Audio recognition using mel spectrograms and convolution neural networks," 2019. [Online]. Available: https://api.semanticscholar.org/CorpusID:237274283

[46] L. L. Wyse, "Audio spectrogram representations for processing with convolutional neural networks," *ArXiv*, vol. abs/1706.09559, 2017. [Online]. Available: https://api.semanticscholar.org/CorpusID:2210510

[47] E. Tsalera, A. Papadakis, and M. Samarakou, "Comparison of pre-trained cnns for audio classification using transfer learning," *Journal of Sensor and Actuator Networks*, vol. 10, no. 4, p. 72, Dec 2021, doi:10.1109/ICASSP43922.2022.9747236.

[48] S. Sudholt and G. A. Fink, "Evaluating word string embeddings and loss functions for cnn-based word spotting," in *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, vol. 01, 2017, pp. 493–498, doi:10.1109/ICDAR.2017.87.

[49] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 37, no. 9, pp. 1904–1916, 2015, doi:https://doi.org/10.1007/978-3-319-10578-$9_2$3.

[50] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, and C. Liu, "A survey on deep transfer learning," in *Artificial Neural Networks and Machine Learning – ICANN 2018*, V. Kůrková, Y. Manolopoulos, B. Hammer, L. Iliadis, and I. Maglogiannis, Eds. Cham: Springer International Publishing, 2018, pp. 270–279, doi:https://doi.org/10.1007/978-3-030-01424-$7_2$7.

[51] Y. Wang, X. Wang, P. Zhu, J. Wu, H. Li, H. Xue, Y. Zhang, L. Xie, and M. Bi, "Opencpop: A High-Quality Open Source Chinese Popular Song Corpus for Singing Voice Synthesis," in *Proc. Interspeech 2022*, 2022, pp. 4242–4246, doi:10.21437/Interspeech.2022-48.

[52] Y. N. Dauphin, A. Fan, M. Auli, and D. Grangier, "Language modeling with gated convolutional networks," in *International conference on machine learning*. PMLR, 2017, pp. 933–941, doi:https://doi.org/10.48550/arXiv.1612.08083.

[53] Y. Lei, S. Yang, X. Wang, and L. Xie, "Msemotts: Multi-scale emotion transfer, prediction, and control for emotional speech synthesis," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 30, pp. 853–864, 2022, doi:10.1109/ISCSLP49672.2021.9362104.

[54] T. Toda, A. W. Black, and K. Tokuda, "Voice conversion based on maximum-likelihood estimation of spectral parameter trajectory," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 8, pp. 2222–2235, 2007, doi:10.1109/TASL.2007.907344.

[55] M. Nishimura, K. Hashimoto, K. Oura, Y. Nankaku, and K. Tokuda, "Singing voice synthesis based on deep neural networks," in *Interspeech*, 2016, pp. 2478–2482, doi:10.21437/Interspeech.2016-1027.

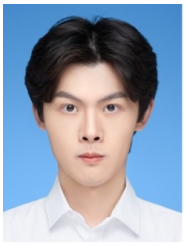
[56] A. Mathews, L. Xie, and X. He, "Semstyle: Learning to generate stylised image captions using unaligned text," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8591–8600, doi:10.1109/CVPR.2018.00896.

[57] J. Wu and J. Luan, "Adversarially Trained Multi-Singer Sequence-to-Sequence Singing Synthesizer," in *Proc. Interspeech 2020*, 2020, pp. 1296–1300, doi:10.21437/Interspeech.2020-1109.

[58] L. Zhang, R. Li, S. Wang, L. Deng, J. Liu, Y. Ren, J. He, R. Huang, J. Zhu, X. Chen, *et al.*, "M4singer: A multi-style, multi-singer and musical score provided mandarin singing corpus," *Advances in Neural Information Processing Systems*, vol. 35, pp. 6914–6926, 2022. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2022/file/2de60892dd329683ec21877a4e7c3091-Paper-Datasets_and_Benchmarks.pdf

[59] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image

recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778, doi:10.1109/CVPR.2016.90.



Junchuan Zhao Junchuan Zhao received his M.Sc. degree in Computer Science from National University of Singapore in Singapore in 2023 and B.Sc. Degree in Telecommunications Engineering with Management from Beijing University of Posts and Telecommunications in China in 2022. He is currently a Ph.D. student at the School of Computing, National University of Singapore, advised by Professor Ye Wang. His research focuses on singing voice synthesis, generative modeling, deep learning, singing voice expression control.

Low Qi Hong Chetwin Low Qi Hong Chetwin is currently pursuing a bachelor's degree in computer science at the National University of Singapore. His research interest includes generative modelling in both audio and computer vision.

Ye Wang Ye Wang is an Associate Professor in the Computer Science Department at the National University of Singapore (NUS). He received his Ph.D. degree from Tampere University of Technology in Finland in 2002, M.Sc. degree from Braunschweig University of Technology in Germany in 1993, and B.Sc. degree from South China University of Technology in China in 1983. He established and directed the sound and music computing (SMC) Lab (https://smcnus.comp.nus.edu.sg). Before joining NUS, he was a member of the technical staff at Nokia Research Center in Tampere, Finland for 9 years. His research philosophy is that technology should be developed for good - such as expanding access, increasing affordability, and improving quality of healthcare and education. Guided by this philosophy, he explored a new programmatic research agenda, which became his signature research in the past decade: cognitive neuroscience-inspired Sound and Music Computing for Human Health and Potential (SMC4HHP), attempting to address two big questions. 1) How to enable users to discover their preferred music that satisfies clinical requirements for Rhythmic Auditory Stimulation (RAS) based gait rehabilitation and exercise via music search, recommendation and generation? 2) How to leverage on the relationship between speech and singing to build applications for speech intervention? To address the above questions, he led the development of MusicRx technologies to make RAS accessible and affordable, and of SLIONS for speech intervention for various populations.