M4Singer: A Multi-Style, Multi-Singer and Musical Score Provided Mandarin Singing Corpus

Lichao Zhang[1], **Ruiqi Li**[1], **Shoutong Wang**[1], **Liqun Deng**[2], **Jinglin Liu**[1], **Yi Ren**[1]
Jinzheng He[1], **Rongjie Huang**[1], **Jieming Zhu**[2], **Xiao Chen**[2], **Zhou Zhao**[1*]
[1]Zhejiang University
{zju_zlc,ruiqili,zhaozhou}@zju.edu.cn
[2]Huawei Noah's Ark Lab
{dengliqun.deng,jamie.zhu}@huawei.com

Abstract

The lack of publicly available high-quality and accurately labeled datasets has long been a major bottleneck for singing voice synthesis (SVS). To tackle this problem, we present **M4Singer**, a free-to-use **M**ulti-style, **M**ulti-singer **M**andarin singing collection with elaborately annotated **M**usical scores as well as its benchmarks. Specifically, 1) we construct and release a large high-quality Chinese singing voice corpus, which is recorded by 20 professional singers, covering 700 Chinese pop songs as well as all the four SATB types (i.e., soprano, alto, tenor, and bass); 2) we take extensive efforts to manually compose the musical scores for each recorded song, which is necessary to the study of the prosody modeling for SVS. 3) To facilitate the use and demonstrate the quality of M4Singer, we conduct four different benchmark experiments: score-based SVS, controllable singing voice (CSV), singing voice conversion (SVC) and automatic music transcription (AMT). Audio samples can be found at http://m4singer.github.io.

1 Introduction

Recently, deep learning-based singing voice synthesis (SVS) has attracted a lot of attention from both industry and academic communities [4; 29; 21; 7; 5; 11; 9; 57; 26; 54; 41; 50]. However, due to high cost of data acquisition, the lack of large and high-quality singing voice corpus for public use has always existed, which heavily hinders the development of the SVS research. Taking the Chinese SVS corpus as the focus, existing open resources suffer from either low quality [38], limited amount [26; 48], or lack of necessary musical score annotation [17]. As a result, it is difficult for researchers to dive into SVS and work out comparable results as the text-to-speech (TTS) community does. To fully facilitate SVS-related studies, for example, score-based SVS, controllable singing voice (CSV), singing voice conversion (SVC), etc., a desirable SVS corpus is supposed to satisfy the following goals:

- *High-quality*: The vocal recordings should be clean enough with high fidelity and professional singing techniques, to guarantee the SVS models generate expressive and assessable synthesized results.
- *Multi-singer*: It is crucial to provide a large size of singing voices from multiple singers, as different sources help enrich the acoustic features to be modeled, which promises the SVC research or zero-shot SVS.

*Corresponding Author

36th Conference on Neural Information Processing Systems (NeurIPS 2022) Track on Datasets and Benchmarks.

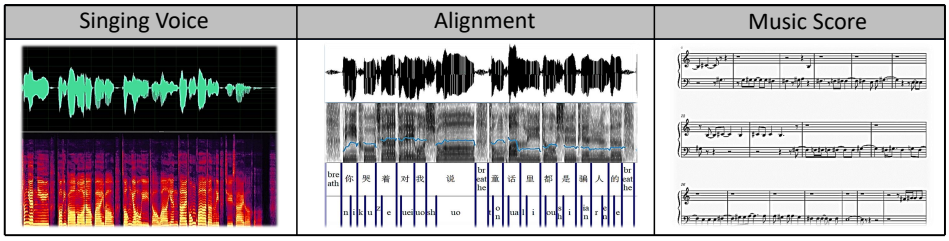

Figure 1: The composition of M4Singer. There are 700 songs recorded by 20 professional singers. Alignment and musical score are manually created by annotators based on each recorded song.

- *Multi-style*: Multiple singing styles bring a wide range of features in pitch and prosody, which are important for expressive SVS, CSV, etc.
- *Rich musical score information*: Unlike TTS which is conditioned on the text scripts only, SVS runs with extra musical score information. Hence, necessary musical scores like note duration, note pitch, etc., should also be accurately prepared.

In this paper, we focus on the Chinese SVS, and make a contribution to the community by releasing **M4Singer**, a large **M**ulti-style, **M**ulti-singer **M**andarin singing collection with elaborately annotated **M**usical scores (see Figure 1). Specifically, M4Singer contains 29.77 hours of high-quality Mandarin singing voice without accompaniment. The songs are recorded by 20 professional singers in a recording studio, covering all four singing styles, i.e., soprano, alto, tenor, and bass (SATB)[2]. More notably, to support pitch modeling based on the musical score, fine-grained phoneme-to-audio alignments as well as manual scores for each recording are also provided.

On the other hand, to facilitate the use of M4Singer, we conduct extensive experiments and build benchmarks on four tasks, i.e., score-based SVS, controllable singing voice, singing voice conversion and automatic music transcription with the recently published approaches. And we also discuss the potential risks and limitations regarding our dataset.

2 Related Work

Recently, benefited from the rapid development of deep learning and the continuous breakthroughs in TTS techniques [47; 23; 52; 37; 35; 36; 18], singing voice synthesis [45; 21; 14; 5; 38; 2; 27; 40; 9; 13; 10; 51; 17; 8; 43; 1; 22; 48; 54; 41; 19; 56; 53; 25] and its related tasks (e.g., SVC [33; 42; 24; 46; 55]) have received extensive attention in both industry and academic communities. Different from TTS, SVS depends on not only the text scripts (i.e., lyrics) but also the musical notes (including note pitch and note duration). Since both the singing and the annotation demand great professional efforts, it leads to much more challenges than that of speech. As a result, it is of the extraordinarily high cost for singing voice corpus construction. This also explains the reason why it lacks of sufficient open resources in SVS. Although there have been many attempts to address this issue as shown in Table 1, they still suffer from either the limited amounts or the lack of necessary musical annotations. The conundrum of data hunger hence has not been resolved in SVS yet.

Returning to the Chinese singing voice datasets, Ren et al [38] mined the singing resources from the web to construct the DeepSinger system. But these singing recordings are originally scraped with accompaniment and difficult to extract the clean vocals, which finally lead to great decline in audio quality for SVS. OpenSinger [17] is a wonderful SVS corpus with a large amount of high quality songs, and rich choices in singers and singing styles. But unfortunately, the alignment information and musical score composing are missed, making it hard to satisfy the score-based SVS which is the most important scenario in practical applications. Recently, Opencpop [48] makes improvements by

[2]The songs in M4Singer cover the pitch range of 35 (B1, 61.7Hz) to 78 (F#5, 740.0Hz) according to the MIDI standard, which could satisfy all the pitch requirements from bass to soprano. The note pitch p can be converted to frequency f by the formula $f = 440 * 2^{(p-69)/12}$.

Table 1: The information table of different datasets. Alignment and Score indicate whether there is manual alignment and musical score annotation respectively.

Corpus	Language	#Hours	#Singers	Alignment	Score
NUS-48E [12]	English	1.91	12	✓	✗
NHSS [39]	English	7	10	✓	✗
JVS-MuSiC [44]	Japanese	2.28	100	✗	✗
Tohoku Kiritan [30]	Japanese	1	1	✓	✓
PopCS [26]	Chinese	5.89	1	✗	✗
OpenSinger [17]	Chinese	50	66	✗	✗
Opencpop [48]	Chinese	5.25	1	✓	✓
M4Singer (Our)	Chinese	29.77	20	✓	✓

providing manually labelled alignment and score annotations. However, Opencpop is a relatively small-scale corpus, which contains nearly 5-hour singing voices of only one female singer. It is still insufficient for multi-singer and multi-style SVS purpose. In this paper, we aim to construct a multi-singer and multi-style Mandarin singing voice corpus with refined manual alignment and musical score annotation to comprehensively address the problems in previous works.

3 Dataset Description

In this section, we will first present the overall of M4Singer by answering some possibly concerned questions, including a brief description, motivation of construction, the instructions for use and the way to obtain the corpus. Then we describe the construction details of M4Singer and provide necessary statistics about M4Singer.

3.1 Overall: QAs about M4Singer

What is M4Singer? Briefly, M4Singer is a large collection of multi-style singing voice recordings and their elaborately labelled alignment and musical score information, which aims to support SVS related studies.

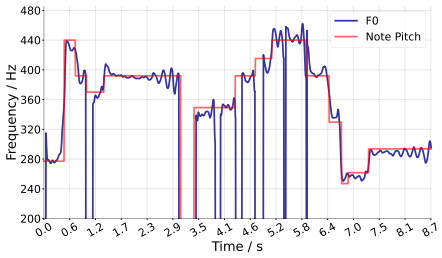

Figure 2: Comparison of F0 and Note Pitch (example comes from M4Singer). Note Pitch is converted to frequency[2].

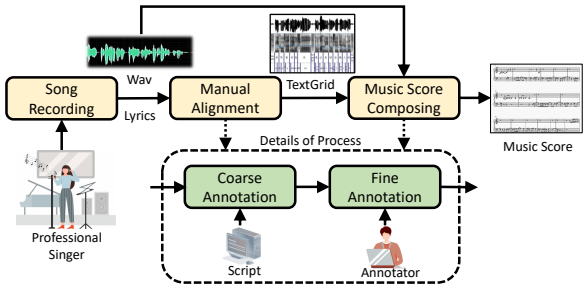

Figure 3: The pipeline of data collection.

Why manual alignment and musical score composing are required? The fundamental frequency (F0) prediction in singing voice synthesis needs to follow the musical score, however, 1) there is a large gap between the note pitch of the score and the fundamental frequency in the actual audio, as shown in Figure 2, which is mainly caused by the vibrato of vocal and the singing skills. Only by providing the musical score corresponding to the singing voice can the model learn to model the prosody from the notes. 2) Besides, although there could be an original musical score (created by the original composer) for each song, it is generally tricky to obtain and even professional singers can

hardly sing fully conformed to the score. 3) And then, given that it is common for pitch shifting while singing, the original musical score is unsuitable for SVS. As a result, we resort to manual alignment and score composing for all recorded songs albeit at a huge cost.

How to get M4Singer? The dataset can be freely downloaded[3] and noncommercially used under license CC BY-NC-SA 4.0. It is worth noting that since our dataset involves sensitive biometric data, i.e., singing voice, even if we desensitize all data, there are still potential risks of singers being doxxed. Therefore, the dataset is only stored on our private servers, and all applications as well as usage will be rigorously reviewed to ensure that the data will not be misused and maliciously spread, for which we ask for your understanding. As for what tasks it is used for, the users can define their own ones under the license in addition to the preset tasks in this paper, but please get in touch with us beforehand. If you have any questions, please contact us by email. Your suggestions will be an important reference for us to maintain our dataset. All the updates will be synced on the website.

3.2 Construction

As shown in Figure 3, the construction of M4Singer is carried out in three steps: song recording, manual alignment, and musical score composing. Among them, manual alignment and musical score composing adopt a combination of batch script and manual work to reduce the difficulty of manual labeling. The *fine annotation* is performed manually after *coarse annotation* processed through batch script. Specific steps are as follows:

Step 1: Song Recording

Table 2: The information table of singers.

Gender	SingerID	#F0	Note Range	#Hours
Female	*Alto-1*	325.76	55 - 72 (G3, 196.0Hz - C5, 523.3Hz)	0.97
	Alto-2	317.8	54 - 72 (F#3, 185.0Hz - C5, 523.3Hz)	0.94
	Alto-3	352.19	54 - 73 (F#3, 185.0Hz - C#5, 554.4Hz)	0.91
	Alto-4	292.16	51 - 72 (D#3, 155.6Hz - C5, 523.3Hz)	1.05
	Alto-5	324.15	52 - 73 (E3, 164.8Hz - C#5, 554.4Hz)	2.63
	Alto-6	301.26	50 - 73 (D3, 146.8Hz - C#5, 554.4Hz)	2.53
	Alto-7	289.22	50 - 72 (D3, 146.8Hz - C5, 523.3Hz)	1.68
	Soprano-1	465.89	61 - 77 (C#4, 277.2Hz - F5, 698.5Hz)	0.88
	Soprano-2	475.63	63 - 77 (D#4, 311.1Hz - F5, 698.5Hz)	0.62
	Soprano-3	513.47	63 - 78 (D#4, 311.1Hz - F#5, 740.0Hz)	1.63
Male	*Tenor-1*	283.47	51 - 69 (D#3, 155.6Hz - A4, 440.0Hz)	1.12
	Tenor-2	222.33	45 - 67 (A2, 110.0Hz - G4, 392.0Hz)	1.15
	Tenor-3	219.07	43 - 67 (G2, 98.0Hz - G4, 392.0Hz)	1.32
	Tenor-4	214.61	45 - 67 (A2, 110.0Hz - G4, 392.0Hz)	1.0
	Tenor-5	177.05	43 - 64 (G2, 98.0Hz - E4, 329.6Hz)	1.76
	Tenor-6	173.86	40 - 63 (E2, 82.4Hz - D#4, 311.1Hz)	1.12
	Tenor-7	178.94	41 - 65 (F2, 87.3Hz - F4, 349.2Hz)	2.41
	Bass-1	109.98	35 - 54 (B1, 61.7Hz - F#3, 185.0Hz)	2.48
	Bass-2	110.93	38 - 55 (D2, 73.4Hz - G3, 196.0Hz)	2.14
	Bass-3	109.72	37 - 51 (C#2, 69.3Hz - D#3, 155.6Hz)	1.43

Singer Selection: We audition a group of professional singers and finally 20 singers who together cover all the full SATB singing styles are selected for the follow-up recordings, including 10 males and 10 females. All singers sign an agreement and agree to open source their singing voice for academic usage before collection. And then considering the task requirements pre-designed and the capability of the singers at the same time, they are asked to sing in a specific pitch range. We group singers based on gender and their vocal range. All four main voice types (soprano, alto, tenor, and bass) are covered in our dataset. As shown in Table 2, singers are numbered anonymously according

[3]Dataset can be found at https://github.com/M4Singer/M4Singer.

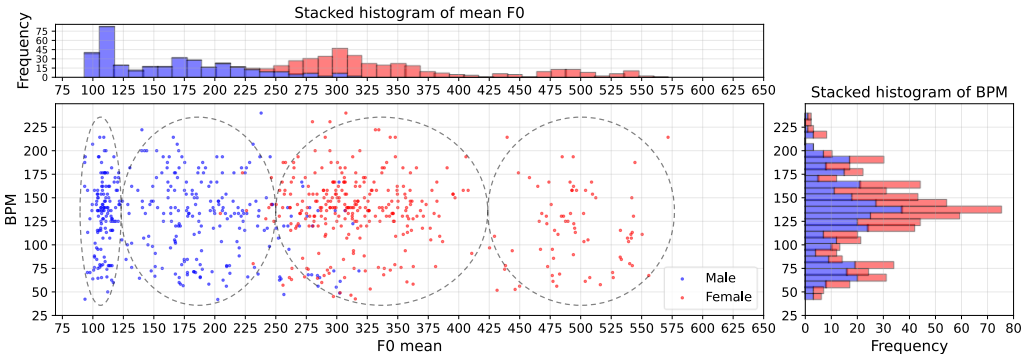

Figure 4: The statistical distribution of the mean F0 and BPM. Each point in the scatter figure represents a song. The top and right figures are stacked histograms of mean F0 and BPM by gender. It can be roughly divided into four groups according to F0 and we draw them with dotted circles, which represent songs of four voice types (bass, tenor, alto, and soprano) from left to right.

to their vocal characteristics. Additionally, the average pitch, note range, and duration of the recorded songs for each singer are demonstrated based on the final recorded 700 songs.

Song Recording: Each singer picks out several Mandarin songs interested in and sings songs within a pre-agreed pitch range as mentioned before. Besides, abundant emotions and certain singing skills (vibrato, portamento, etc.) are also attached while singing. Moreover, the audio only contains unaccompanied vocals recorded in a professional recording studio. All the audio is recorded in 44,100 Hz sampling rate with 24 bits per sample in wav format.

To demonstrate the diversity of the recorded songs intuitively, as shown in Figure 4, we present the mean F0 and BPM of these 700 songs in the form of a scatter plot, and make statistics in the form of a stacked histogram, in which blue and red represent songs of male and female respectively. It can be seen that 1) the songs are clustered in four groups in the scatter plot: 90Hz - 125Hz, 125Hz - 250Hz, 250Hz - 425Hz and 425Hz - 575Hz, which correspond to songs of bass, tenor, alto and soprano respectively. 2) The distribution of F0 (top) has a clear relationship with gender, in which 250Hz can be used as an approximate boundary. In contrast, the distribution of BPM (right) has little correlation with gender, in which the BPM of the songs are basically gathered at 75 - 175. 3) The wide range of singers and the wide range of F0 and BPM demonstrate that the system trained with M4Singer could sustain a large variety of pitch, rate, and timbre conditions potentially.

Step 2: Manual Alignment

Coarse Annotation: We leverage open source tools to process the audio and lyrics firstly, 1) Pypinyin[4] is used to convert Chinese lyrics to phonemes. 2) Montreal forced alignment (MFA) [28] is used to align original lyrics and audio. The results of coarse annotation are kept in the form of TextGrid.

Fine Annotation: Through the *coarse annotation*, rough boundaries of words and phonemes can be obtained. Annotators leverage Praat [6] to make corrections on the rough annotation results, mainly concentrating on the following aspects: 1) Boundary correction: the annotators correct the boundaries of words and phonemes by listening to the audio and observing the spectrogram, which constitutes the main effort of this step. 2) Word correction: to tickle the case of missing or incorrect lyrics, the annotators are required to correct the lyrics based on the audio. It is worth noting that the singers may sing words that do not match the provided lyrics due to the occasional mistakes or accents. Annotators are required to replace them with homophonic words according to the audio. 3) Phoneme correction: according to the *"Scheme for the Chinese Phonetic Alphabet*[5]*"*, the annotators correct the automatically converted phonemes to standard phonemes. On the other hand, due to the existence of

[4]https://github.com/mozillazg/python-pinyin

[5]The original 1958 Scheme, apparently scanned from a reprinted copy in Xinhua Zidian. PDF version from the Chinese Ministry of Education. Click here to download.

polyphonic characters (a word with more than one pronunciation) in Chinese, the phonemes generated in coarse may not match the actual pronunciation of the singer, so annotators must correct them according to what they hear. 4) Unvoiced labeling: the unvoiced region includes breathing and silent sections, and the annotators mark the boundaries respectively. In addition, occasional electrical and non-singing sounds are marked with "noise".

Step 3: Musical Score Composing

Coarse Annotation: Fine word boundaries can be obtained from the previous step. In this step, we first apply the open source tool[6] to extract F0. F0 is then averaged according to the word boundaries to get rough notes and is saved in MIDI form as the score to be annotated. In addition, a fine-grained MIDI is obtained according to the F0 curve as the reference for the determination of note pitch.

Fine Annotation: Twenty annotators with professional musical literacy are recruited for the fine annotation of the musical score. They first drag the two MIDI files obtained by *coarse annotation* into the self-developed software for annotation. And then the annotators focus on note correction and glissando annotation: 1) referring to fine-grained MIDI, the annotators determine the pitch of the notes so that the annotated pitch is in harmony with the singing voice; 2) given that the singers may change the keys while singing, especially when a word is in a long duration, annotators are asked to split the rough note properly and determine the note pitch.

3.3 Post-Processing

Data Check: To ensure a high quality of the annotation, we ask 5 additional professionals to check 25% of the labeled data after annotation, including alignment check and score check. An alignment check includes word and phoneme boundaries, polyphonic words, etc. For the score check, the testers are supposed to put the score and singing voice into the playback software at the same time and estimate whether the notes match the singing voice.

Data Segmentation: After the annotation, we segment the audio into smaller fragments to facilitate training, while the alignment and score are correspondingly segmented into sentence-level segments. By leveraging the manual alignment results, we set a threshold for the unvoiced region as the primary condition for performing the segmentation process. And the voiced region is also constrained with the maximum length as the secondary condition. By doing so, almost all sentences are between 3 and 10 seconds. Finally, 20,942 utterances with a total duration of around 29.77 hours are obtained.

3.4 Statistics

As introduced in Section 3.2, we have statistics on singers and recorded vocals. In this section, we perform statistics on the annotated phonemes as shown in Figure 5, and notes as shown in Figure 6.

Chinese phonemes are divided into initials and finals. An initial is a consonant that precedes a final to form a complete syllable. As mentioned above, the phonemes are entirely annotated according to the *"Scheme for the Chinese Phonetic Alphabet[5]"*, which contains 21 initials and 35 finals. As can be seen, all 21 initials are covered in M4Singer, and even the lowest-occurring phoneme "p" has 2,093 occurrences, which is sufficient for modeling. As for the finals, 34 finals appear in M4Singer, and the lowest-occurring "er" has 496 occurrences. The missing phoneme is "ueng", which is barely seen in Chinese songs, and the sum usage frequency of all words containing it in daily life is less than 0.01% [7], so we are not trying to do special treatment. In addition, we count the annotated note pitch by gender. As shown in figure 6, the note range is about 35 (B1, 61.7Hz) to 78 (F#5, 740.0Hz). And among them, male singers mainly gather in 40 (E2, 82.4Hz) to 60 (C4, 261.6Hz), while female singers mainly gather in 55 (G3, 196.0Hz) to 75 (D#5, 622.3Hz).

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

[7]https://lingua.mtsu.edu/chinese-computing/statistics/char/list.php

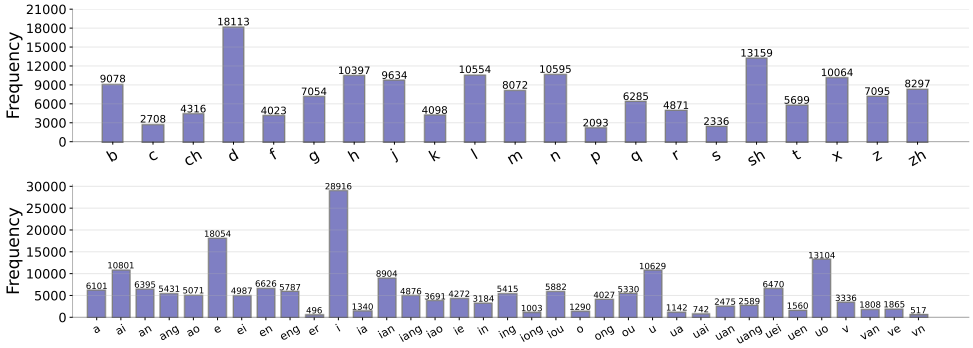

Figure 5: The distribution of phonemes, divided into initials (top) and finals (bottom).

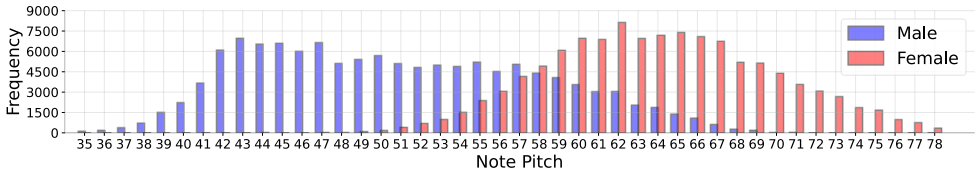

Figure 6: The statistical distribution of pitch by gender. Pitch is presented as MIDI note number[2].

4 Benchmarks

In this section, we first present benchmarks for three SVS-related tasks, i.e., score-based SVS, controllable singing voice (CSV), and singing voice conversion (SVC) to respectively demonstrate music-score, multi-style, and multi-singer features of M4Singer. And then to explore more possibilities, we also perform the task of music information retrieval (MIR), i.e., note-level automatic music transcription (AMT) on M4Singer. The experiments are conducted on a single NVIDIA 2080Ti GPU.

4.1 Score-Based SVS

Previous singing acoustic models are limited by datasets and cannot synthesize singing through musical scores. Therefore, we perform score-based SVS as the first part of benchmarks, and re-evaluate the acoustic model of previous works, mainly in terms of prosody modeling. We select 4 singers considering the singing style, technique, pitch, etc., and then pick up two songs of each test singer as the test songs. Finally, 8 songs of 4 singers are used as the test set, and the rest are used as the training set. All the systems adopt HiFi-GAN [20] as the vocoder.

We conduct experiments on the following systems: 1) *GT*, the ground truth singing audio; 2) *GT (voc.)*, where we first convert the ground truth singing audio to the ground truth mel-spectrograms and then convert these mel-spectrograms back to audio using vocoder; 3) *FFT-Singer*, the SVS system which generates mel-spectrograms through stacked feed-forward Transformer (FFT) blocks and then uses the vocoder to synthesize audio; 4) *GAN-Singer*, the SVS system with adversarial training using multiple random window discriminators; 5) *DiffSinger*, the novel SVS system based on diffusion model. Then we denote the input ground truth F0 by *w/ GT F0* as the upper bound for the prosody modeling of each model. To evaluate the perceptual audio quality, we conduct the MOS (mean opinion score) evaluation on the test set. Among them, MOS-Q indicates the quality of the audio and MOS-P indicates the coherence and naturalness of prosody. These indicators are all scored on a scale of 1 to 5, with higher scores indicating higher quality of the referred explanation. Twenty qualified listeners are asked to make judgments about the synthesized song samples.

The subjective and objective evaluations are shown in Table 3. We find that 1) the generative models (GAN, diffusion, etc.) can greatly improve the sound quality (MOS-Q), just as described in previous works [9; 26]; 2) Although the generative model is also helpful for the modeling of singing techniques

Table 3: The audio performance (MOS-Q and MOS-P), F0-MAE and voiced/unvoiced (V/UV) accuracy comparisons, where both F0 and V/UV are obtained from the mel-spectrogram through a pretrained pitch extractor.

ID	Method	MOS-Q (↑)	MOS-P (↑)	F0-MAE (↓)	V/UV Acc. (↑)
1	*GT*	4.52 ± 0.08	4.47 ± 0.07	/	/
2	*GT (voc.)*	4.26 ± 0.09	4.39 ± 0.09	/	/
3	*FFT-Singer*	3.52 ± 0.14	3.57 ± 0.13	8.1011	0.9314
4	*w/ GT F0*	3.56 ± 0.11	3.75 ± 0.12	1.9321	0.9651
5	*GAN-Singer*	3.68 ± 0.12	3.62 ± 0.13	8.5263	0.9317
6	*w/ GT F0*	3.75 ± 0.09	3.89 ± 0.12	2.1545	0.9723
7	*Diffsinger*	3.92 ± 0.13	3.72 ± 0.15	8.9990	0.9406
8	*w/ GT F0*	3.97 ± 0.10	4.02 ± 0.11	1.7061	0.9856

(vibrato and portamento, etc.) and could generate more diverse patterns of pitch curves, there occur problems of excessive trembling or out-of-tune, resulting in the indicator for prosody is not as high as expected. At the same time, the increase of F0-MAE also indicates that the generated prosody deviates from the real curve, resulting in *prosody oscillation*. 3) When using *GT F0*, the improvement of prosody is far greater than the sound quality. This indicates that there is still a certain room for improvements in prosody compared to the high level of audio quality already achieved. Apart from the issue of the generative model, we consider that it may be due to the large gap between musical score and prosody, and it may be alleviated by converting coarse-grained notes into fine-grained F0 through an intermediate prosody module (the difference between note pitch and F0 can be intuitively grasped through Figure 2) or by auxiliary methods to further constrain the generation process.

4.2 Controllable Singing Voice

In addition to score-based SVS, we consider designing another task to explicitly leverage the multi-style feature provided by M4Singer. Therefore, in this section we introduce the novel task of controllable singing voice (CSV) that enables the prosody modification of existing singing voice based on a given modified musical score while keeping the content and timbre. We first pre-train an ASR model to extract the phonetic posteriorgrams (PPG) of the singing voice. Then keep the ASR encoder and connect it to the acoustic model, in which we use WaveNet [31]. The system takes the original audio and the modified notes as the input and outputs the modified audio.

Method	MOS-P	MOS-S
Unmodified	4.39 ± 0.09	/
F0-Based	3.51 ± 0.12	3.58 ± 0.11
Score-Based	3.63 ± 0.15	3.62 ± 0.13

Table 4: Comparison between different pitch shifting systems. *Score-Based* means shifting through the score. *F0-Based* means shifting through F0. MOS-S to measure the similarity of timbre. *Unmodified* means the original audio through the vocoder.

SV \ TV	*Bass*	*Tenor*	*Alto*	*Soprano*
Bass	0	-0.03	-0.09	-1.24
Tenor	-0.07	0	-0.04	-0.12
Alto	-0.19	-0.05	0	-0.07
Soprano	-1.43	-0.27	-0.03	0

Table 5: The audio performance (CMOS) of shifting between different voice types. CMOS (↑) is scaled of -2 to 2. "SV" denotes the source voice type. "TV" denotes the target voice type.

We implement two sets of experiments in this task. In the first set, a random pitch change from -12 to 12 is conducted on each test sentence. As shown in Table 4, the *score-based* pitch shifting outperforms *F0-based*, especially in the naturalness of prosody, which manifests in pitch variability and vibrato performance. Then we conduct the second experiment on the mutual shifting among the four singing voice types respectively. The shift of adjacent voice types changes 6 pitches (e.g., from bass to tenor, it adds 6 pitches, and from bass to alto changes 12 pitches). Additionally, the shift between the same type remains unchanged and is used as the reference for evaluation. As shown in Figure 5, the shifting between adjacent types exhibits fine performance on all four types, revealing

superior controllability of the model. However, CMOS decreases rapidly as the number of spanning types increases, and even severe distortion appears when four types are spanned.

4.3 Singing Voice Conversion

Singing voice conversion (SVC) aims to convert the voice of one singer to that of other singers while keeping the singing content and prosody [24]. Unlike previous works, here we use musical scores as input for prosody modeling rather than F0. We reuse the Diffsinger model in section 4.1, but reset the dataset partition to conduct the sing adaptation task on M4Singer. A female and a male singer are left as target singers and the rest as source singers. All songs of source singers are first used to train the base model, and then the base model is fine-tuned by training on 15 minutes of audio randomly selected from each target singer. During inference, the model takes scores and lyrics that do not belong to the target singer as well as the timbre of the target singer to synthesize the singing voice to validate the adaptation performance of the model. Moreover, We apply a simple pitch shift method to the musical score to alleviate the pitch bias between different singers, where the reference audio is used to correct the input notes.

Table 6: The performance of audio on prosody naturalness (MOS-P) and timbre similarity (MOS-S) in the singing voice conversion. "TG" denotes the target gender, who is the provider of timbre. "SG" denotes the source gender, who is the provider of score and lyrics. "PS" denotes the pitch shift.

TG	SG	*w/o PS*		*w/ PS*	
		MOS-P	MOS-S	MOS-P	MOS-S
F	**F**	3.67 ± 0.11	3.76 ± 0.13	3.74 ± 0.13	3.79 ± 0.09
	M	3.12 ± 0.16	3.19 ± 0.15	3.67 ± 0.12	3.76 ± 0.10
M	**M**	3.58 ± 0.13	3.72 ± 0.13	3.61 ± 0.12	3.74 ± 0.11
	F	3.16 ± 0.17	3.45 ± 0.12	3.65 ± 0.09	3.73 ± 0.14

As shown in Table 6, We conduct the experiments by gender, where the naturalness of prosody (MOS-P) and the similarity (MOS-S) to the original timbre concerned in this task is demonstrated. The results show that M4Singer can reach a high score in SVC. And pitch shift contributes to a performance uplift both in naturalness and similarity especially when the score-provider and the target singer are of different gender indicating that musical score is also singer-dependent, and it would be a good direction to explore how to adapt score outside the singer's domain.

4.4 Automatic Music Transcription

Note-level automatic music transcription (AMT) refers to converting a recorded music piece into its symbolic form containing the onset, offset, and pitch of every note [3; 32]. We use VOCANO [16; 58; 49] as the baseline model. MIR-1K [15] is also used as the semi-supervised dataset to assist virtual adversarial training. We randomly sample 8 hours of the M4Singer corpus (2 hours for each voice type) and train the model on it. The performance is then validated on a test set with 2 hours data (0.5 hours for each voice type), which is randomly selected from the remaining part of M4Singer (Row 1 in Table 7). We measure all the scores using utilities in the `mir_eval` library [34] with onset tolerance set to 100ms. The experiment can be regarded as two different tasks, i.e., onset-offset detection and note transcription. The former focuses more on the boundaries of notes, and the latter requires more attention on pitch prediction.

We then explore the performances for different voice types. A clear trend indicates that the performance increases from bass to tenor, and then decreases a little for alto and soprano (Row 2-5). A poor performance appears when it comes to too low or too high frequencies. The reduction of performance is particularly significant on note transcription because VOCANO potentially degenerates to degrade performance when predicting pitch of bass and soprano.

Table 7: Results of automatic music transcription baselines. The model is trained on an 8-hour subset of M4Singer, where the sizes for each voice type are balanced. Row 1 is the results of testing on the 2-hour test subset of M4Singer. Row 2-5 are the partial results of testing on four 0.5-hour test subsets of specific voice types.

ID	Test Set	F1 (onset)	F1 (offset)	Precision (note)	Recall (note)	F1 (note)
1	M4Singer-AMT-Test	0.7794	0.8206	0.5589	0.4693	0.5060
2	Bass (Subset)	0.8100	0.8243	0.4768	0.4034	0.4338
3	Tenor (Subset)	0.7961	0.8367	0.5991	0.5201	0.5334
4	Alto (Subset)	0.7828	0.8212	0.5690	0.4838	0.5191
5	Soprano (Subset)	0.7264	0.8003	0.5979	0.4760	0.5242

5 Limitations and Conclusion

In this paper, we delve into the shortcomings of the current singing corpora and propose a novel corpus M4Singer, a multi-style, multi-singer, musical score provided Mandarin singing corpus to comprehensively tickle the problems existing before. The construction process and statistical information of M4Singer are presented in detail. And finally, we preset benchmarks for M4Singer on four different tasks to provide a reference for subsequent research.

We claim three limitations of M4Singer. First, although the labeling process is carried out by professionals with musical backgrounds, affected by the subjectivity of hearing, there are inevitably slight deviations in the perception of duration and pitch. Then, long-tailed distribution still exists in note pitches, even though we have tried to cover the full range, which may have an impact on the performance of model in boundary cases. Finally, as we discussed above, the corpus is at risk because it contains sensitive biometric data. We will strictly implement qualification and maintenance terms to ensure the rational use of the corpus.

Acknowledgement

This work was supported in part by the National Natural Science Foundation of China (Grant No.62072397 and No.61836002), Zhejiang Natural Science Foundation (LR19F020006), Yiwise, and National Key R&D Program of China (Grant No.2020YFC0832505). We appreciate the support from Mindspore: https://www.mindspore.cn, which is a new deep learning computing framework.

References

[1] J. Alonso and C. Erkut. Latent space explorations of singing voice synthesis using ddsp. *ArXiv*, abs/2103.07197, 2021.

[2] O. Angelini, A. Moinet, K. Yanagisawa, and T. Drugman. Singing synthesis: with a little help from my attention. In *INTERSPEECH*, 2020.

[3] E. Benetos, S. Dixon, D. Giannoulis, H. Kirchhoff, and A. Klapuri. Automatic music transcription: challenges and future directions. *intelligent information systems*, 2013.

[4] M. Blaauw and J. Bonada. A neural parametric singing synthesizer modeling timbre and expression from natural songs. *Applied Sciences*, 7:1313, 2017.

[5] M. Blaauw and J. Bonada. 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, 2020.

[6] P. Boersma. Praat, a system for doing phonetics by computer. *Glot International*, 2002.

[7] P. Chandna, M. Blaauw, J. Bonada, and E. Gómez. Wgansing: A multi-voice singing voice synthesizer based on the wasserstein-gan. *2019 27th European Signal Processing Conference (EUSIPCO)*, pages 1–5, 2019.

[8] F. Chen, R. Huang, C. Cui, Y. Ren, J. Liu, and Z. Zhao. Singgan: Generative adversarial network for high-fidelity singing voice generation. *arXiv preprint arXiv:2110.07468*, 2021.

[9] 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.

[10] 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)*, pages 319–323. IEEE, 2021.

[11] S. Choi, W. Kim, S. Park, S. Yong, and J. Nam. Korean singing voice synthesis based on auto-regressive boundary equilibrium gan. *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7234–7238, 2020.

[12] Z. Duan, H. Fang, B. Li, K. C. Sim, and Y. Wang. The nus sung and spoken lyrics corpus: A quantitative comparison of singing and speech. *asia pacific signal and information processing association annual summit and conference*, 2013.

[13] 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)*, pages 1–5. IEEE, 2021.

[14] Y. Hono, K. Hashimoto, K. Oura, Y. Nankaku, and K. Tokuda. Singing voice synthesis based on generative adversarial networks. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6955–6959. IEEE, 2019.

[15] C.-L. Hsu and J.-S. R. Jang. On the improvement of singing voice separation for monaural recordings using the mir-1k dataset. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(2):310–319, 2010.

[16] J.-Y. Hsu and L. Su. VOCANO: A note transcription framework for singing voice in polyphonic music. In *Proc. International Society of Music Information Retrieval Conference (ISMIR)*, 2021.

[17] R. Huang, F. Chen, Y. Ren, J. Liu, C. Cui, and Z. Zhao. Multi-singer: Fast multi-singer singing voice vocoder with a large-scale corpus. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 3945–3954, 2021.

[18] R. Huang, M. W. Lam, J. Wang, D. Su, D. Yu, Y. Ren, and Z. Zhao. Fastdiff: A fast conditional diffusion model for high-quality speech synthesis. *arXiv preprint arXiv:2204.09934*, 2022.

[19] S. Kim, K. Na, C. Lee, J. An, and I. Kim. U-singer: Multi-singer singing voice synthesizer that controls emotional intensity. *arXiv preprint arXiv:2203.00931*, 2022.

[20] J. Kong, J. Kim, and J. Bae. Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis. *neural information processing systems*, 2020.

[21] J. Lee, H.-S. Choi, C.-B. Jeon, J. Koo, and K. Lee. Adversarially trained end-to-end korean singing voice synthesis system. *arXiv preprint arXiv:1908.01919*, 2019.

[22] 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.

[23] N. Li, S. Liu, Y. Liu, S. Zhao, M. Liu, and M. Zhou. Neural speech synthesis with transformer network. *arXiv: Computation and Language*, 2018.

[24] Z. Li, B. Tang, X. Yin, Y. Wan, L. Xu, C. Shen, and Z. Ma. Ppg-based singing voice conversion with adversarial representation learning. *international conference on acoustics, speech, and signal processing*, 2020.

[25] C.-F. Liao, J.-Y. Liu, and Y.-H. Yang. Karasinger: Score-free singing voice synthesis with vq-vae using mel-spectrograms. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 956–960. IEEE, 2022.

[26] J. Liu, C. Li, Y. Ren, F. Chen, P. Liu, and Z. Zhao. Diffsinger: Singing voice synthesis via shallow diffusion mechanism. *arXiv preprint arXiv:2105.02446*, 2, 2021.

[27] P. Lu, J. Wu, J. Luan, X. Tan, and L. Zhou. Xiaoicesing: A high-quality and integrated singing voice synthesis system. 2020.

[28] M. McAuliffe, M. Socolof, S. Mihuc, M. Wagner, and M. Sonderegger. Montreal forced aligner: Trainable text-speech alignment using kaldi. In *Interspeech*, volume 2017, pages 498–502, 2017.

[29] K. Nakamura, K. Hashimoto, K. Oura, Y. Nankaku, and K. Tokuda. Singing voice synthesis based on convolutional neural networks. *arXiv preprint arXiv:1904.06868*, 2019.

[30] I. Ogawa and M. Morise. Tohoku kiritan singing database: A singing database for statistical parametric singing synthesis using japanese pop songs. *Acoustical Science and Technology*, 2021.

[31] A. v. d. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016.

[32] M. Pesek, A. Leonardis, and M. Marolt. Robust real-time music transcription with a compositional hierarchical model. *PLOS ONE*, 2017.

[33] A. Polyak, L. Wolf, Y. Adi, and Y. Taigman. Unsupervised cross-domain singing voice conversion. In *INTERSPEECH*, 2020.

[34] C. Raffel, B. Mcfee, E. Humphrey, J. Salamon, O. Nieto, D. Liang, and D. Ellis. mir_eval: A transparent implementation of common mir metrics. In *Proceedings - 15th International Society for Music Information Retrieval Conference (ISMIR 2014)*, 10 2014.

[35] Y. Ren, C. Hu, X. Tan, T. Qin, S. Zhao, Z. Zhao, and T.-Y. Liu. Fastspeech 2: Fast and high-quality end-to-end text to speech. *arXiv preprint arXiv:2006.04558*, 2020.

[36] Y. Ren, J. Liu, and Z. Zhao. Portaspeech: Portable and high-quality generative text-to-speech. *arXiv: Audio and Speech Processing*, 2021.

[37] Y. Ren, Y. Ruan, X. Tan, T. Qin, S. Zhao, Z. Zhao, and T.-Y. Liu. Fastspeech: fast, robust and controllable text to speech. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, pages 3171–3180, 2019.

[38] 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*, pages 1979–1989, 2020.

[39] B. Sharma, X. Gao, K. Vijayan, X. Tian, and H. Li. Nhss: A speech and singing parallel database. *Speech Communication*, 2020.

[40] J. Shi, S. Guo, N. Huo, Y. Zhang, and Q. Jin. Sequence-to-sequence singing voice synthesis with perceptual entropy loss. *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 76–80, 2021.

[41] J. Shi, S. Guo, T. Qian, N. Huo, T. Hayashi, Y. Wu, F. Xu, X. Chang, H. Li, P. Wu, et al. Muskits: an end-to-end music processing toolkit for singing voice synthesis. *arXiv preprint arXiv:2205.04029*, 2022.

[42] B. Sisman and H. Li. Generative adversarial networks for singing voice conversion with and without parallel data. *The Speaker and Language Recognition Workshop (Odyssey 2020)*, 2020.

[43] J. Tae, H. Kim, and Y. Lee. Mlp singer: Towards rapid parallel korean singing voice synthesis. *2021 IEEE 31st International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6, 2021.

[44] H. Tamaru, S. Takamichi, N. Tanji, and H. Saruwatari. Jvs-music: Japanese multispeaker singing-voice corpus. *(CD-ROM)*, 2020.

[45] 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*, 32:55–73, 2015.

[46] C. Wang, Z. Li, B. Tang, X. Yin, Y. Wan, Y. Yu, and Z. Ma. Towards high-fidelity singing voice conversion with acoustic reference and contrastive predictive coding. 2022.

[47] Y. Wang, R. Skerry-Ryan, D. Stanton, Y. Wu, R. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio, Q. V. Le, Y. Agiomyrgiannakis, R. A. J. Clark, and R. A. Saurous. Tacotron: Towards end-to-end speech synthesis. *conference of the international speech communication association*, 2017.

[48] 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. *arXiv preprint arXiv:2201.07429*, 2022.

[49] Y.-T. Wu, Y.-J. Luo, T.-P. Chen, I.-C. Wei, J.-Y. Hsu, Y.-C. Chuang, and L. Su. Omnizart: A general toolbox for automatic music transcription. *arXiv preprint arXiv:2106.00497*, 2021.

[50] H. Xue, X. Wang, Y. Zhang, L. Xie, P. Zhu, and M. Bi. Learn2sing 2.0: Diffusion and mutual information-based target speaker svs by learning from singing teacher. *arXiv preprint arXiv:2203.16408*, 2022.

[51] F.-R. Yang, Y.-P. Cho, Y.-H. Yang, D.-Y. Wu, S.-H. Wu, and Y.-W. Liu. Mandarin singing voice synthesis with a phonology-based duration model. In *2021 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pages 1975–1981. IEEE, 2021.

[52] C. Yu, H. Lu, N. Hu, M. Yu, C. Weng, K. Xu, P. Liu, D. Tuo, S. Kang, G. Lei, D. Su, and D. Yu. Durian: Duration informed attention network for multimodal synthesis. *arXiv: Computation and Language*, 2019.

[53] X. Zhang, J. Wang, N. Cheng, and J. Xiao. Susing: Su-net for singing voice synthesis. *arXiv preprint arXiv:2205.11841*, 2022.

[54] Y. Zhang, J. Cong, H. Xue, L. Xie, P. Zhu, and M. Bi. Visinger: Variational inference with adversarial learning for end-to-end singing voice synthesis. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7237–7241. IEEE, 2022.

[55] Y. Zhang, P. Yang, J. Xiao, Y. Bai, H. Che, and X. Wang. K-converter: An unsupervised singing voice conversion system. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6662–6666. IEEE, 2022.

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

[57] X. Zhuang, T. Jiang, S.-Y. Chou, B. Wu, P. Hu, and S. Lui. Litesing: Towards fast, lightweight and expressive singing voice synthesis. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7078–7082. IEEE, 2021.

[58] F. Zih-Sing and L. Su. Hierarchical classification networks for singing voice segmentation and transcription. In *ISMIR*, 2019.