STARS: A Unified Framework for Singing Transcription, Alignment, and Refined Style Annotation

Wenxiang Guo† Yu Zhang† Changhao Pan† Zhiyuan Zhu
Ruiqi Li Zhetao Chen Wenhao Xu Fei Wu * Zhou Zhao *
Zhejiang University
{guowx314,yuzhang34,zhaozhou}@zju.edu.cn

Abstract

Recent breakthroughs in singing voice synthesis (SVS) have heightened the demand for high-quality annotated datasets, yet manual annotation remains prohibitively labor-intensive and resource-intensive. Existing automatic singing annotation (ASA) methods, however, primarily tackle isolated aspects of the annotation pipeline. To address this fundamental challenge, we present STARS, which is, to our knowledge, the first unified framework that simultaneously addresses singing transcription, alignment, and refined style annotation. Our framework delivers comprehensive multi-level annotations encompassing: (1) precise phoneme-audio alignment, (2) robust note transcription and temporal localization, (3) expressive vocal technique identification, and (4) global stylistic characterization including emotion and pace. The proposed architecture employs hierarchical acoustic feature processing across frame, word, phoneme, note, and sentence levels. The novel non-autoregressive local acoustic encoders enable structured hierarchical representation learning. Experimental validation confirms the framework's superior performance across multiple evaluation dimensions compared to existing annotation approaches. Furthermore, applications in SVS training demonstrate that models utilizing STARS-annotated data achieve significantly enhanced perceptual naturalness and precise style control. This work not only overcomes critical scalability challenges in the creation of singing datasets but also pioneers new methodologies for controllable singing voice synthesis. Audio samples are available at https://demo-stars.github.io.

1 Introduction

Automatic Singing Annotation (ASA) constitutes the computational process of extracting key vocal features from singing recordings, encompassing phonetic transcriptions (phoneme alignment), MIDI note parameters (pitch/duration), and stylistic attributes (emotion/technique). As the cornerstone of modern singing voice synthesis (SVS) systems, ASA provides fine-grained, multi-level annotated data for training expressive and controllable singing synthesis models. While recent breakthroughs in generative models and controllable SVS frameworks (Guo et al., 2025; Zhang et al., 2024a,b) have dramatically improved the quality of generated singing voices, they have also paradoxically exposed a critical bottleneck: the scarcity of high-quality annotated singing corpora.

Traditional annotation workflows require labor-intensive manual processing by audio engineers and musicians, making large-scale dataset creation both costly and time-consuming. While some open-source singing datasets such as OpenCpop (Wang et al., 2022b) and VocalSet (Wilkins et al., 2018) have attempted to alleviate this burden, their annotations are limited to basic phonetic or vocal technique information. Recent datasets like GTSinger (Zhang et al., 2024c) have made significant progress by incorporating a wider range of annotations—from basic phoneme and note annotations to various singing techniques and global styles. However, the volume of data remains insufficient compared to the scale of speech corpora.

Modern SVS systems require multi-level annotation precision across four key dimensions: (1) microsecond-aligned phoneme boundaries for prosody modeling; (2) accurate MIDI note timing/pitch for melody preservation; (3) phone-level vocal technique recognition (e.g., vibrato, falsetto); and (4) global stylistic attributes (emotion, pace). As shown in Figure 1, traditional solutions employ fragmented toolchains—combining ASR systems like WhisperX (Bain et al., 2023) and Qwen-Audio (Chu et al., 2024) for lyric transcription, MFA (McAuliffe et al., 2017) for forced alignment,

[†] Equal contribution
[*] Corresponding Author

Findings of the Association for Computational Linguistics: ACL 2025, pages 15081–15093
July 27 - August 1, 2025 ©2025 Association for Computational Linguistics

and pitch trackers like VOCANO (Hsu et al., 2021) and MusicYOLO (Wang et al., 2022a). This patchwork approach introduces cascading errors from tool mismatches while failing to capture expressive vocal styles. Such disjointed processes hinder the creation of large, high-quality annotated singing datasets necessary for cutting-edge SVS models.

To overcome these challenges, we propose STARS, a unified framework for multi-level singing voice annotation that streamlines the entire process. STARS offers three key innovations: (1) A multi-level architecture for extracting singing information at various granularities, covering frame, word, phoneme, note, and sentence levels; (2) A local acoustic encoder that works together with a CMU encoder employing a U-Net architecture with Conformer blocks and FreqMOE, and with a vector quantization, to extract acoustic features at multiple hierarchical levels; and (3) A multi-task automated annotation pipeline that sequentially predicts phoneme boundaries, note boundaries, pitch, phone-level techniques, and global stylistic attributes. We demonstrate the effectiveness of STARS through comprehensive evaluations across multiple ASA tasks. Our framework achieves superior performance in phoneme alignment accuracy and note prediction precision. When applied to SVS model training, STARS-annotated data yields significant improvements in synthesized voice naturalness and style control accuracy.

- We propose STARS, the first unified framework for Singing Transcription, Alignment, and Refined Style Annotation.
- We design a five-level unified architecture with specialized acoustic encoders for hierarchical feature filtering and extraction.
- We implement parallel prediction strategies for phoneme/note boundaries and pitch estimation, enhanced through phone-level technique and global style detection.
- Through training SVS models with our annotations, we demonstrate the practical utility of STARS in achieving superior singing vocal naturalness and precise style control.

2 Related Works

2.1 Singing Voice Synthesis

Singing Voice Synthesis (SVS) aims to generate expressive singing voices from musical scores. Early models such as XiaoiceSing (Lu et al., 2020) adopt non-autoregressive acoustic architectures inspired by FastSpeech (Ren et al., 2019). Subsequent work, ViSinger (Zhang et al., 2022), employs VITS (Kim et al., 2021) to establish an end-to-end SVS framework. Generative adversarial networks (Wu and Luan, 2020; Huang et al., 2022b) and diffusion models (Liu et al., 2022) have also been applied to enhance audio fidelity. Controllable SVS focuses on manipulating vocal attributes including timbre, emotion, and style to achieve more expressive performances. Resna and Rajan (2023) develops a multi-singer framework for cross-voice synthesis. Muse-SVS (Kim et al., 2023) enables precise control over pitch, energy, and phoneme duration to express different emotional intensities. Prompt-Singer (Wang et al., 2024) introduces natural language prompts to achieve fine-grained control over singing voices. To mitigate data scarcity, DeepSinger (Ren et al., 2020) constructs a large-scale corpus by mining singing data from online sources. Additionally, OpenCpop (Wang et al., 2022b) and GTSinger (Zhang et al., 2024c) provide publicly available corpora with manually annotated singing recordings. Nevertheless, the limited availability of high-quality singing data remains a critical bottleneck compared to speech resources.

2.2 Automatic Singing Annotation

Automatic Singing Annotation (ASA) includes tasks such as lyric alignment, note estimation and segmentation, and vocal technique and style annotation. The MFA (McAuliffe et al., 2017) is a conventional approach for lyric alignment. However, singing voice alignment remains challenging due to the large variations in phoneme durations and rhythmic structures. Several studies (Wang et al., 2023; Huang et al., 2022a) adopt Viterbi forced alignment (Forney, 1973) to improve the accuracy of aligning posteriograms with lyrics. For note estimation, VOCANO (Hsu et al., 2021) and MusicYOLO (Wang et al., 2022a) directly predict the pitch and duration of musical notes. ROSVOT (Li et al., 2024) incorporates phoneme boundary priors to achieve MIDI note prediction. SongTrans (Wu et al., 2024) builds upon the Whisper model and adopts a hybrid autoregressive and non-autoregressive framework for phoneme and note annotation. MusCaps (Manco et al., 2021) leverages a language model to generate music captions, but its effectiveness is limited for singing voice recordings without background music. Despite these advancements, exist-

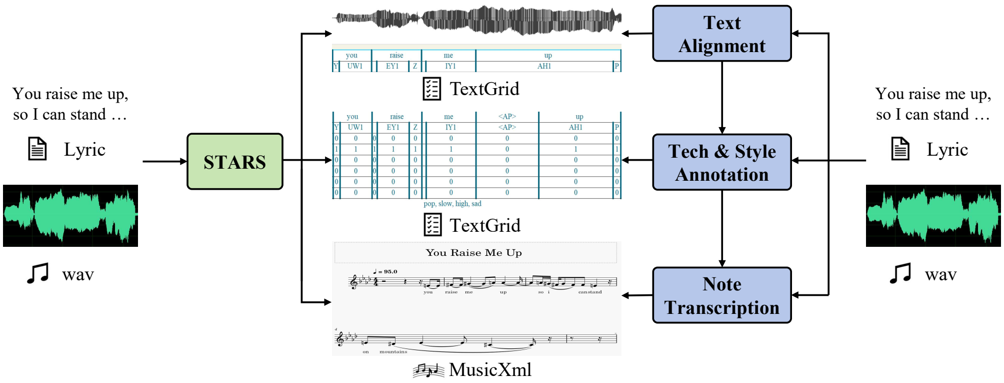


Figure 1: STARS vs. Traditional Stepwise Pipeline. Conventional stepwise processing requires sequential execution of text alignment, note transcription, and manual technique and style annotation, with error propagation across cascaded modules. STARS establishes unified acoustic-linguistic modeling for simultaneous phoneme, MIDI, technique, and style prediction, eliminating error accumulation via end-to-end joint optimization.

ing ASA methods remain fragmented, requiring separate models and manual integration.

3 STARS

3.1 Problem Formulation

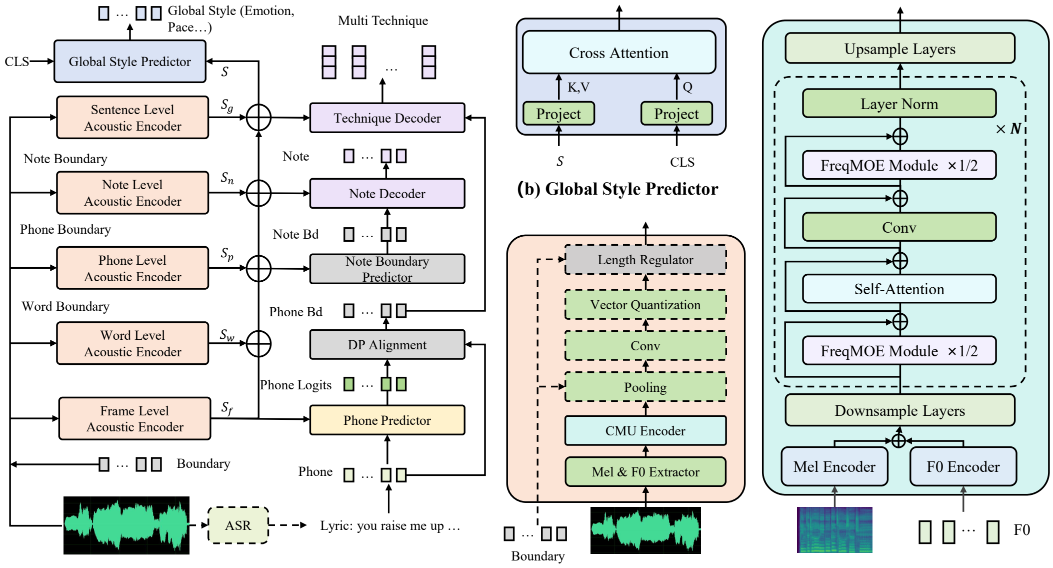
Typically, a Mel-spectrogram $\mathbf{M} \in \mathbb{R}^{T \times F}$ is derived from the audio signal using the short-time Fourier transform (STFT), where T denotes the number of frames and F the number of frequency bins. Let the phoneme sequence be represented as $p = [p_1, p_2, \ldots, p_{L_p}]$, where L_p is the number of phonemes. In addition to predicting the phoneme sequence, the model is required to predict the corresponding boundaries for each phoneme, represented as $ph\_bd = [pbd_1, pbd_2, \ldots, pbd_T]$, where $pbd_i = 1$ indicates that frame i is a phoneme boundary and 0 otherwise. Furthermore, for each phoneme, the model predicts a set of singing techniques. For each technique $i \in \{1, \ldots, 9\}$, a binary sequence $tech^i = [t_1^i, t_2^i, \ldots, t_{L_p}^i]$ is predicted, where $t_j^i = 1$ indicates the presence of technique i at the j-th phoneme and $t_j^i = 0$ indicates its absence. Based on the predicted phoneme and word boundaries, note boundaries are predicted as $note\_bd = [nbd_1, nbd_2, \ldots, nbd_T]$. The model also predicts the note pitch sequence $c = [c_1, c_2, \ldots, c_{L_n}]$, where L_n denotes the number of notes. In addition to the phoneme, note, and technique predictions, the model is required to predict global sentence-level attributes $\mathbf{g}$.

3.2 Overview

Figure 2 illustrates the overall architecture of STARS. The input to our system consists of Mel-spectrograms and F0 extracted from the audio signal, and the corresponding lyrics can be obtained using ASR models such as Whisper (Radford et al., 2023). To improve the robustness of our model, we follow the approach of ROSVOT (Li et al., 2024) by adding realistic noise from the MUSAN dataset (Snyder et al., 2015) to the input audio and injecting Gaussian noise into the extracted F0 contour. In this section, we first describe the unified multi-level framework. To capture multi-level acoustic and stylistic information, we design a hierarchical architecture that spans five levels: Frame, Word, Phone, Note, and Sentence. Each level shares the same backbone while employing slightly different methods to efficiently extract features at varying granularities. Next, we explain how all sub-tasks are completed in a single forward pass. To obtain the Phone and Word boundaries, we first predict the frame-level phoneme logits from the features extracted by the Frame-level encoder. We then apply Viterbi forced alignment to determine the phoneme and word boundaries. For note boundaries, we utilize features from the previous three levels to predict the note boundaries. Having extracted features at the Note level, we can predict the corresponding note pitch. Finally, leveraging the information from all five levels, we predict various singing techniques and global attributes.

3.3 Unified Multi-Level Framework

To achieve unified annotation predictions at multiple levels within a single model, we design a Unified Multi-Level Framework consisting of five hierarchical levels: Frame, Word, Phone, Note, and Sentence. Each level extracts acoustic features at



(a) Overall Architecture **(c) Multi-Level Acoustic Encoder** **(d) CMU Encoder**

Figure 2: The overall architecture of STARS. (a) The unified multi-level framework, which integrates singing lyric alignment, note transcription, technique prediction, and global style prediction. (b) The Global Style Predictor, where [CLS] tokens are used as queries, and S represents the keys and values. (c) The multi-level acoustic encoder extracts features at each level, with optional pooling based on boundary segmentation. (d) The CMU Encoder, employing a U-Net architecture with Conformer blocks and FreqMOE for efficient audio feature extraction.

different granularities. The framework employs a shared acoustic encoder across all levels to enable efficient feature extraction. We first design a highly efficient CMU Encoder to extract features, followed by multi-granularity pooling operations. Vector quantization serves as a bottleneck (Van Den Oord et al., 2017) to eliminate irrelevant information. We then use the boundary to expand the extracted features to the frame length T.

The CMU Encoder module utilizes a U-Net-based architecture for Mel-spectrogram downsampling while preserving spectral details through skip connections during upsampling. To capture both long-term and short-term dependencies in the time dimension, we integrate the Conformer architecture (Gulati et al., 2020), which demonstrates superior performance in the ASR tasks. For enhanced frequency analysis, we design the FreqMOE module that partitions the frequency dimension into K equal frequency bands and applies the specialized experts to distinct frequency bands. We then concatenate the output embedding chunks. Specific details are provided in the Appendix A.2.

For the pooling module, no pooling is applied to the frame-level features, as this level represents the finest granularity. For each intermediate level, we utilize the corresponding boundary information to perform segmentation, followed by average pooling within these boundaries to obtain dynamic feature sequences. At the sentence level, we apply global average pooling to the CMU Encoder outputs to obtain the holistic representation.

For hierarchical representation learning, we apply vector quantization to the intermediate features from all levels except at the Frame and Sentence levels. Let $\mathbf{S}_l \in \mathbb{R}^{L\times D}$ denote the input latent embeddings for level l, where L is the sequence length and D is the feature dimension. Each level maintains a codebook $\mathbf{q}_l \in \mathbb{R}^{K\times D}$ containing K latent embeddings. Following (Van Den Oord et al., 2017), we also apply a commitment loss to ensure that the representation sequence commits to an embedding and to prevent the output from growing:

$$\mathcal{L}_{\text{commit}} = \|\mathbf{z}_l(\mathbf{S}_l) - \text{sg}[\mathbf{q}_l]\|_2^2, \tag{1}$$

where $z_l(.)$ is the vector quantization module for level l, and sg denotes the stop gradient operator.

In the Length Regulator module, at each level $l \in \{\text{word}, \text{phone}, \text{note}\}$, we use the boundary information to determine the frame length of each segment. To align hierarchical representations, we repeat each embedding according to the length of the segment. At the Sentence level, we extend the single embedding to match the frame length T.

3.4 Lyric Alignment

Phonemes are the smallest phonetic units and the most commonly used tokens in singing voice synthesis. To align phonemes with the audio, we input the frame-level extracted features, denoted as $\mathcal{S}_f$, into the phone predictor. This predictor generates predictions for the phoneme being sung at each frame, along with indications of whether a frame corresponds to a phoneme boundary. During training, we optimize the phone predictions using cross-entropy loss, and we also apply Connectionist Temporal Classification (CTC) loss (Graves et al., 2006) to further improve phoneme alignment.

$$L_{\text{CE}} = -\sum_{k=1}^{P} y_k \log(p_k), \tag{2}$$

$$\mathcal{L}_{\text{CTC}} = -\log \sum_{\pi \in \mathcal{B}^{-1}(\mathbf{y})} P(\pi|\mathbf{X}), \tag{3}$$

where P is the total number of phonemes, and $\pi = [\pi_1, \ldots, \pi_T]$ represents an alignment path with $\pi_t \in \mathcal{V} \cup \{\text{blank}\}$ (where $\mathcal{V}$ is the total phoneme vocabulary). Additionally, we use BCE loss to predict the phone boundaries.

During inference, we employ the Viterbi forced alignment (Forney, 1973) method to align the lyric phoneme sequence with the phone probability distribution at each frame. This process provides the phoneme boundaries, and through the phone-to-word relationship, we further determine the word boundaries. Specific details of the inference algorithms are provided in the Appendix B.

3.5 Note Transcription and Alignment

To obtain note boundaries, we first fuse features from the frame, word, and phone levels and feed them into a note boundary predictor. The training loss for note boundary prediction is defined as the BCE loss, similar to phone boundary prediction. During inference, since the word boundaries overlap with the note boundaries, we employ these boundaries as constraints when generating the final note boundaries. For pitch prediction at the note level, we integrate the aggregated features from the frame, word, and phone levels with note-level features. Specifically, for each note j, we denote its feature sequence as $\mathbf{S}_n^j \in \mathbb{R}^{L_j \times D}$, where L_j is the number of frames in the note segment and D is the feature dimension. We then use the note decoder to predict the pitch of each note. Inspired by CIF (Dong and Xu, 2020), we also compute the weight vector for each note as $W_n = \mathbf{S}_n^j \mathbf{W}_a$, where $\mathbf{W}_a$ is a learnable projection matrix and $W_n \in \mathbb{R}^{L_j \times 1}$. We then obtain the aggregated note representation c_j via a weighted average:

$$c_j = \sum_{t=1}^{L_j} \mathbf{w}_n(t)\, \mathbf{S}_n^j(t), \tag{4}$$

where $\mathbf{w}_n(t)$ denotes the weight at frame t.

We then use the pitch predictor to compute the logits for the P pitch categories $\hat{p}_j = c_j \mathbf{W}_O$, where $\mathbf{W}_O \in \mathbb{R}^{D \times P}$. The pitch predictor is also optimized using cross-entropy loss.

3.6 Technique and Global Style Predictor

To predict the possible techniques for each phoneme, we treat this task as a multi-task, multi-label binary classification problem. The technique prediction head outputs predictions across nine categories: mixed, falsetto, strong, weak, glissando, breathy, bubble, vibrato, and pharyngeal. For each phoneme j, we predict the i-th technique $tech_j^i$, where 1 indicates the presence of the technique and 0 indicates its absence. Features from the Frame, Word, Phone, Note, and Sentence levels are input into the model, with the previously obtained phone boundaries used as references. We apply the same attention-based weighted average strategy used for pitch prediction to predict each technique sequence. Binary cross-entropy (BCE) loss is used to optimize each technique prediction module.

For the global attributes of the sentence, including language, gender, emotion, pace, and range, we treat these as multi-class classification tasks. To predict these global attributes, we introduce five [CLS] tokens, each corresponding to one of the tasks. These tokens are used as queries $\mathcal{H}_c$, while the sum of the frame-level features from all levels, i.e., $\mathcal{S}_f$, $\mathcal{S}_w$, $\mathcal{S}_p$, $\mathcal{S}_n$, and $\mathcal{S}_g$, serves as the key and value in the cross-attention mechanism (Vaswani, 2017). Positional encoding embeddings are added to the features to determine the position of each token. The formulation is as follows:

$$\begin{gathered}\mathcal{S} = \mathcal{S}_f + \mathcal{S}_w + \mathcal{S}_p + \mathcal{S}_n + \mathcal{S}_g, \\ \text{Attention}(\mathcal{H}_c, \mathcal{S}, \mathcal{S}) = \text{Softmax}\left(\frac{\mathcal{H}_c \mathcal{S}^T}{\sqrt{D}}\right)\mathcal{S},\end{gathered} \tag{5}$$

where D is the dimension of the query, key and value. Then, we predict each task's category $g_i \in \mathbb{R}^{C_i}$, where C_i is the number of categories for the i-th attribute. Cross-entropy (CE) loss is used to train and optimize the global style predictors.

3.7 Training and Inference Procedures

During training, we leverage ground truth phone, word, and note boundaries to guide the model. The final loss function consists of the following components: 1) L_{ph}: A combination of phone-level CTC loss and CE loss; 2) L_{pbd} and L_{nbd}: Boundary prediction losses for the phone and note, respectively, optimized using BCE loss; 3) L_{pi}: Pitch prediction loss, calculated as the CE loss between the ground truth pitch and the predicted pitch; 4) L_{tech}: Technique prediction loss, calculated as the CE loss; 5) L_g: The global attribute prediction loss, optimized using CE loss; and 6) L_c: The commitment loss, which constrains the vector quantization layer.

During inference, we first obtain the phonemes and words from the input lyrics (or ASR model-generated lyrics) and use the frame-level predicted phoneme logits with Viterbi forced alignment to determine the phoneme boundaries. Note boundaries are then predicted by feeding fused features from the frame, word, and phone levels into the note boundary predictor. The note-level features, together with the fused features, are provided as input to the note-level acoustic encoder and the note decoder to predict the note pitch. Next, features from all levels, along with the phoneme boundaries, are used to predict the techniques for each phoneme. Finally, the global attributes are predicted using the aggregated features from all levels.

4 Experiments

4.1 Experimental Setup

4.1.1 Dataset and Process

The dataset used in our experiments includes the Chinese and English subsets of GTSinger (Zhang et al., 2024c). This dataset provides alignments and annotations in TextGrid files, which include word boundaries, phoneme boundaries, phoneme-level annotations for six techniques (mixed, falsetto, pharyngeal, glissando, vibrato, breathy), and global style labels such as emotion, pace, and pitch range. Additionally, we have collected and annotated a 30-hour Chinese dataset featuring two singers and four technique annotations (mixed, falsetto, strong, weak, breathy, bubble) at both the phoneme and sentence levels. To train our automated annotation model, we reserve 30 songs containing various techniques and global styles as the validation and test sets. To ensure the robustness of the model, we augment the dataset by adding noise from the MUSAN noise corpus (Snyder et al., 2015). For Chinese lyrics, we use the pypinyin tool[1] to phonemize the text, while for English lyrics, we follow the ARPA[2] standard for phoneme transcription.

4.1.2 Implementation Details

The singing audio recordings are sampled at 24 kHz, with a window size of 512 samples, a hop size of 128, and 80 mel bins for Mel-spectrogram extraction. We use a pre-trained RMVPE (Wei et al., 2023) model to extract the F0 contours. The U-Net backbone consists of four downsampling and upsampling layers, with a total downsampling factor of 16×. The Conformer module includes two layers, and the FreqMOE consists of four experts. In this experiment, the model is trained for 150k steps using an NVIDIA 4090 GPU. Further implementation details are provided in Appendix A.1.

4.1.3 Evaluation Details

For lyric alignment, we evaluate performance using two metrics: Boundary Error Rate (BER) and Intersection Over Union (IOU) score. BER measures the proportion of misplaced boundaries within 20ms tolerance. The IOU score is defined as the ratio of the duration of the overlapping segment between two notes to the duration of the combined time span covered by both notes. For note transcription, we use the `mir_eval` library (Raffel et al., 2014) and apply the metrics COnPOff (correct onset, pitch, and offset) proposed in (Molina et al., 2014) and Raw Pitch Accuracy (RPA) for overall note pitch prediction performance. For phone-level technique and global style recognition, we use objective metrics including F1 score and accuracy to evaluate the phone-level technique predictor, and accuracy for the global style detector. The results are multiplied by 100 for better readability. Further details are provided in Appendix C.2.

4.1.4 Baseline Models

To evaluate our approach, we compare it with several baseline systems across different sub-tasks. We conduct the comparison using only Chinese data, and additionally, we test the model's performance

[1] https://github.com/mozillazg/python-pinyin
[2] https://en.wikipedia.org/wiki/ARPABET

Method	BER ↓	IOU ↑
MFA	40.3	56.8
SOFA	20.9	80.0
STARS (ours)	**18.6**	**80.9**

Table 1: Results for lyric alignment.

Method	COnPOff(F) ↑	RPA ↑
VOCANO	50.2	76.6
ROSVOT	70.2	83.8
STARS (ours)	**71.0**	**86.7**

Table 2: Results for note transcription and alignment.

Setting	Metric	Phone-level Technique and Global Style Prediction										
		BUB	BRE	PHA	VIB	GLI	MIX	FAL	WEA	STR	**TEC**	**STY**
GTSinger	F1	46.9	68.7	88.7	95.7	78.5	61.5	33.2	37.2	82.5	67.3	-
	ACC	31.5	73.2	75.7	99.3	78.9	93.9	40.8	17.4	95.3	65.9	-
STARS	F1	71.7	66.9	85.0	65.5	72.3	74.7	93.5	90.3	99.4	**79.9**	-
	ACC	97.8	88.8	95.4	96.7	84.1	81.9	94.7	90.4	93.9	**91.5**	68.0

Table 3: The objective results of phone-level technique prediction. The singing techniques include BUB (bubble), BRE (breathy), PHA (pharyngeal), VIB (vibrato), GLI (glissando), MIX (mixed), FAL (falsetto), WEA (weak), and STR (strong). The "TEC" column represents the average metrics calculated across all the singing techniques, while the "STY" column represents the average metrics calculated across all the global style attributes.

on multilingual data by combining both Chinese and English datasets. For the phoneme and singing audio alignment, we consider: 1) Montreal Forced Align (MFA): a tool that aligns orthographic and phonological forms by leveraging a pronunciation dictionary to time-align transcribed audio files; 2) SOFA [3]: a forced alignment tool designed specifically for the singing voice. For note alignment and transcription, we compare our model's performance with the results reported in ROSVOT, including two strong baselines:1) VOCANO: a note transcription framework developed for the singing voice in polyphonic music; 2) ROSVOT: a model that employs a multi-scale architecture for automatic singing transcription. We compare with the variant without word boundary condition. For phone technique prediction, we use GTSinger's technique predictor as the baseline.

4.2 ASA Results

4.2.1 Lyric Alignment and Note Transcription

In Table 1, we observe that for the lyric alignment task, when comparing our model with the other two models, STARS achieves the best performance in both the BER and IOU metrics, indicating its ability to accurately predict phoneme boundary information. In Table 2, we see that for the note alignment and transcription task, our model outperforms the baseline models in both COnPOff(F) and RPA metrics, demonstrating its sensitivity to note boundaries and pitch. Notably, while the other models are designed to handle only a single task, our model efficiently handles both lyric alignment and note transcription tasks simultaneously. This demonstrates the versatility and effectiveness of STARS in performing multiple singing annotation tasks, making it highly suitable for foundational automatic singing annotation applications.

4.2.2 Technique and Global Style Prediction

As Table 3 shows, for the recognition of the nine vocal techniques, our experimental results outperform GTSinger. No individual technique shows a significantly low recognition accuracy. The average F1-score and accuracy for all techniques far exceed the GTSinger benchmark, demonstrating our model's ability to accurately detect and annotate multiple techniques at the phoneme level. Also, as the last column of the table indicates, our model achieves high accuracy in recognizing global style attributes, further showcasing its effectiveness in capturing the overall stylistic features of singing audio. For detailed scores of each style attribute, see the Appendix C.3. In summary, our model effectively detects expressive information, and the generated labels can be used in various controllable expressive singing voice synthesis tasks.

4.2.3 Ablation Study

In this section, we conduct ablation experiments to evaluate the contributions of different components in our model. We test the following variants: 1) w/o CTC: the model without the CTC loss; 2) w/o VQ: the model without vector quantization; 3) w/o

[3] https://github.com/qiuqiao/SOFA

Method	BER ↓	IOU ↑	COnPOff(F) ↑	RPA ↑	T-F1 ↑	T-ACC ↑	S-ACC ↑
STARS	18.6	**80.9**	**71.0**	**86.7**	**79.9**	**91.5**	68.0
w/o CTC	19.1	80.0	70.9	86.7	79.7	89.8	66.8
w/o VQ	**18.3**	80.9	70.7	86.4	76.3	90.4	65.3
w/o MOE	18.9	80.5	70.1	86.5	77.4	90.0	69.8
Conv	20.1	78.3	66.4	82.7	62.9	89.6	66.6
Bilingual	19.4	75.6	68.1	86.2	76.8	91.4	**70.4**

Table 4: The ablation results for different sub-tasks. T-F1 is the average F1-score for all singing techniques, T-ACC is the average accuracy for all singing techniques, and S-ACC is the average accuracy for all global style attributes.

Train	Infer	MOS-Q ↑	MOS-C ↑
GT	**GT**	3.98 ± 0.09	4.05 ± 0.09
GT	**Pred**	3.91 ± 0.07	3.95 ± 0.08
Pred	**Pred**	3.89 ± 0.11	3.95 ± 0.05
1/2 GT	**GT**	3.83 ± 0.04	3.89 ± 0.10
Mix	**Mix**	3.93 ± 0.06	3.98 ± 0.05

Table 5: Results of SVS. GT refers to the ground truth, Pred indicates the results from our ASA model, and Mix represents a mix where half of the data is automatically annotated and the other half is ground truth.

MOE: the model without the FreqMOE strategy; 4) Conv: the model with the Conformer module replaced by a convolutional architecture; 5) Bilingual: the model trained on bilingual datasets.

As shown in Table 4, several conclusions can be drawn. Comparing the first row with the w/o CTC variant, we observe improvements of 0.5 and 0.9 in the BER and IOU metrics for lyric alignment, respectively, along with enhanced performance in the technique recognition task, which is sensitive to phoneme boundaries, while the note recognition metrics remain similar. These results indicate that CTC loss notably enhances phoneme alignment. When VQ is omitted for the phone, note, and word levels, the note, technique, and style recognition tasks show improved performance. For phoneme alignment, the lighter model with reduced VQ loss training further improves phoneme boundary detection. Comparisons of the w/o MOE and Conv variants reveal a drop in performance across all metrics, demonstrating the effectiveness of Finally, experiments on bilingual datasets confirm that our model operates effectively in a multilingual setting. Detailed results for single-language and bilingual experiments can be found in Appendix C.3.

4.3 Singing Voice Synthesis

To further validate STARS's effectiveness, we use STARS to annotate GTSinger's Chinese part and our data, and employ the latest style-controllable SVS model TCSinger(Zhang et al., 2024b) for generation tasks. We use MOS-Q for quality and naturalness assessment, and MOS-C for expressiveness evaluation of the generated singing's style control.

As shown in Table 5, the following conclusions can be drawn: By comparing the first two rows, where training is performed on ground-truth, we observe that using our model's annotations yields MOS-C and MOS-Q scores nearly identical to those obtained with ground-truth annotations. Similarly, comparing the second and third rows, which correspond to training with ground-truth versus our model's annotations, and evaluating with our predicted results, shows only minimal differences in MOS-C and MOS-Q scores. These findings indicate that training exclusively with our annotated data achieves performance comparable to using ground-truth annotations, demonstrating the effectiveness of our fully automated annotation model. Furthermore, comparing the last two rows—where training is conducted with half ground-truth data versus a mixture of half ground-truth and half predicted annotations—reveals an improvement in performance. This suggests that augmenting the dataset with our model's predictions can effectively enhance overall SVS model performance.

5 Conclusion

In this paper, we introduce STARS, the first unified framework for Singing Transcription, Alignment, and Refined Style Annotation. We construct a multi-level framework that efficiently extracts audio features at five granularities—frame, word, phone, note, and sentence—using a hierarchical acoustic encoder. Our approach enables a complete automatic singing annotation pipeline, sequentially performing singing Lyric alignment, note transcription and alignment, phone-level technique prediction, and global style prediction. Experimental results demonstrate that our model achieves high performance across all sub-tasks, and the annotated outputs are further validated in a singing synthesis task, confirming the effectiveness of our approach.

6 Limitations

Our model has two main limitations. First, it currently only classifies global style attributes of the singing voice and generates simple captions using predefined templates. In the future, we aim to enhance this capability by connecting the model to large language models, enabling more dynamic and context-aware caption generation. Second, the model has been validated solely on Chinese and English datasets. Further validation on datasets from other languages is necessary to assess its generalizability. In future work, we plan to extend the model to singing data in additional languages.

7 Ethics Statement

The automatic singing annotation model may be subject to misuse in the following ways. The model could be used to generate synthetic singing voices that closely resemble real individuals or artists without their consent, potentially leading to concerns around authenticity and intellectual property. As the model is trained primarily on English and Chinese datasets, its performance on other languages or diverse cultural contexts may be limited, potentially resulting in biased or inaccurate annotations for non-target languages or dialects. To mitigate these issues, we encourage transparent usage, proper attribution, and the continued development of ethical guidelines for synthetic media generation. Additionally, we also plan to explore the ways to make the model more inclusive and adaptable to a broader range of languages and contexts.

Acknowledgement

This work was supported by National Natural Science Foundation of China under Grant No.62222211 and National Natural Science Foundation of China under Grant No.U24A20326.

References

Max Bain, Jaesung Huh, Tengda Han, and Andrew Zisserman. 2023. Whisperx: Time-accurate speech transcription of long-form audio. *arXiv preprint arXiv:2303.00747*.

Yunfei Chu, Jin Xu, Qian Yang, Haojie Wei, Xipin Wei, Zhifang Guo, Yichong Leng, Yuanjun Lv, Jinzheng He, Junyang Lin, et al. 2024. Qwen2-audio technical report. *arXiv preprint arXiv:2407.10759*.

Linhao Dong and Bo Xu. 2020. Cif: Continuous integrate-and-fire for end-to-end speech recognition. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6079–6083. IEEE.

G David Forney. 1973. The viterbi algorithm. *Proceedings of the IEEE*, 61(3):268–278.

Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. 2006. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning*, pages 369–376.

Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, et al. 2020. Conformer: Convolution-augmented transformer for speech recognition. *arXiv preprint arXiv:2005.08100*.

Wenxiang Guo, Yu Zhang, Changhao Pan, Rongjie Huang, Li Tang, Ruiqi Li, Zhiqing Hong, Yongqi Wang, and Zhou Zhao. 2025. Techsinger: Technique controllable multilingual singing voice synthesis via flow matching. *arXiv preprint arXiv:2502.12572*.

Jui-Yang Hsu, Li Su, et al. 2021. Vocano: A note transcription framework for singing voice in polyphonic music.

Jiawen Huang, Emmanouil Benetos, and Sebastian Ewert. 2022a. Improving lyrics alignment through joint pitch detection. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 451–455. IEEE.

Rongjie Huang, Chenye Cui, Feiyang Chen, Yi Ren, Jinglin Liu, Zhou Zhao, Baoxing Huai, and Zhefeng Wang. 2022b. Singgan: Generative adversarial network for high-fidelity singing voice generation. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 2525–2535.

Jaehyeon Kim, Jungil Kong, and Juhee Son. 2021. Conditional variational autoencoder with adversarial learning for end-to-end text-to-speech. In *International Conference on Machine Learning*, pages 5530–5540. PMLR.

Sungjae Kim, Yewon Kim, Jewoo Jun, and Injung Kim. 2023. Muse-svs: Multi-singer emotional singing voice synthesizer that controls emotional intensity. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*.

Ruiqi Li, Yu Zhang, Yongqi Wang, Zhiqing Hong, Rongjie Huang, and Zhou Zhao. 2024. Robust singing voice transcription serves synthesis. *Preprint*, arXiv:2405.09940.

Jinglin Liu, Chengxi Li, Yi Ren, Feiyang Chen, and Zhou Zhao. 2022. Diffsinger: Singing voice synthesis via shallow diffusion mechanism. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pages 11020–11028.

Peiling Lu, Jie Wu, Jian Luan, Xu Tan, and Li Zhou. 2020. Xiaoicesing: A high-quality and integrated singing voice synthesis system. *arXiv preprint arXiv:2006.06261*.

Ilaria Manco, Emmanouil Benetos, Elio Quinton, and György Fazekas. 2021. Muscaps: Generating captions for music audio. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8.

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

Emilio Molina, Ana Maria Barbancho-Perez, Lorenzo Jose Tardon-Garcia, Isabel Barbancho-Perez, et al. 2014. Evaluation framework for automatic singing transcription.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2023. Robust speech recognition via large-scale weak supervision. In *International conference on machine learning*, pages 28492–28518. PMLR.

Colin Raffel, Brian McFee, Eric J Humphrey, Justin Salamon, Oriol Nieto, Dawen Liang, Daniel PW Ellis, and C Colin Raffel. 2014. Mir_eval: A transparent implementation of common mir metrics. In *ISMIR*, volume 10, page 2014.

Yi Ren, Yangjun Ruan, Xu Tan, Tao Qin, Sheng Zhao, Zhou Zhao, and Tie-Yan Liu. 2019. Fastspeech: Fast, robust and controllable text to speech. *Advances in neural information processing systems*, 32.

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

S Resna and Rajeev Rajan. 2023. Multi-voice singing synthesis from lyrics. *Circuits, Systems, and Signal Processing*, 42(1):307–321.

David Snyder, Guoguo Chen, and Daniel Povey. 2015. Musan: A music, speech, and noise corpus. *arXiv preprint arXiv:1510.08484*.

Aaron Van Den Oord, Oriol Vinyals, et al. 2017. Neural discrete representation learning. *Advances in neural information processing systems*, 30.

A Vaswani. 2017. Attention is all you need. *Advances in Neural Information Processing Systems*.

Jun-You Wang, Chon-In Leong, Yu-Chen Lin, Li Su, and Jyh-Shing Roger Jang. 2023. Adapting pretrained speech model for mandarin lyrics transcription and alignment. In *2023 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 1–8. IEEE.

Xianke Wang, Bowen Tian, Weiming Yang, Wei Xu, and Wenqing Cheng. 2022a. Musicyolo: A vision-based framework for automatic singing transcription. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 31:229–241.

Yongqi Wang, Ruofan Hu, Rongjie Huang, Zhiqing Hong, Ruiqi Li, Wenrui Liu, Fuming You, Tao Jin, and Zhou Zhao. 2024. Prompt-singer: Controllable singing-voice-synthesis with natural language prompt. *arXiv preprint arXiv:2403.11780*.

Yu Wang, Xinsheng Wang, Pengcheng Zhu, Jie Wu, Hanzhao Li, Heyang Xue, Yongmao Zhang, Lei Xie, and Mengxiao Bi. 2022b. Opencpop: A high-quality open source chinese popular song corpus for singing voice synthesis. *arXiv preprint arXiv:2201.07429*.

Haojie Wei, Xueke Cao, Tangpeng Dan, and Yueguo Chen. 2023. Rmvpe: A robust model for vocal pitch estimation in polyphonic music. *arXiv preprint arXiv:2306.15412*.

Julia Wilkins, Prem Seetharaman, Alison Wahl, and Bryan Pardo. 2018. Vocalset: A singing voice dataset. In *ISMIR*, pages 468–474.

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

Siwei Wu, Jinzheng He, Ruibin Yuan, Haojie Wei, Xipin Wei, Chenghua Lin, Jin Xu, and Junyang Lin. 2024. Songtrans: An unified song transcription and alignment method for lyrics and notes. *arXiv preprint arXiv:2409.14619*.

Yongmao Zhang, Jian Cong, Heyang Xue, Lei Xie, Pengcheng Zhu, and Mengxiao Bi. 2022. 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.

Yu Zhang, Wenxiang Guo, Changhao Pan, Zhiyuan Zhu, Tao Jin, and Zhou Zhao. 2025a. Isdrama: Immersive spatial drama generation through multimodal prompting. *arXiv preprint arXiv:2504.20630*.

Yu Zhang, Wenxiang Guo, Changhao Pan, Zhiyuan Zhu, Ruiqi Li, Jingyu Lu, Rongjie Huang, Ruiyuan Zhang, Zhiqing Hong, Ziyue Jiang, et al. 2025b. Versatile framework for song generation with prompt-based control. *arXiv preprint arXiv:2504.19062*.

Yu Zhang, Rongjie Huang, Ruiqi Li, JinZheng He, Yan Xia, Feiyang Chen, Xinyu Duan, Baoxing Huai, and

Zhou Zhao. 2024a. Stylesinger: Style transfer for out-of-domain singing voice synthesis. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 19597–19605.

Yu Zhang, Ziyue Jiang, Ruiqi Li, Changhao Pan, Jinzheng He, Rongjie Huang, Chuxin Wang, and Zhou Zhao. 2024b. Tcsinger: Zero-shot singing voice synthesis with style transfer and multi-level style control. *arXiv preprint arXiv:2409.15977*.

Yu Zhang, Changhao Pan, Wenxiang Guo, Ruiqi Li, Zhiyuan Zhu, Jialei Wang, Wenhao Xu, Jingyu Lu, Zhiqing Hong, Chuxin Wang, et al. 2024c. Gtsinger: A global multi-technique singing corpus with realistic music scores for all singing tasks. *arXiv preprint arXiv:2409.13832*.

A Details of Model

A.1 Architecture

The model input for STARS consists of Mel spectrograms and the extracted f0, which are encoded separately using two encoders, denoted as $\mathrm{E_M}$ for the Mel spectrogram and $\mathrm{E_P}$ for f0. The Mel encoder is constructed from linear layers and residual convolution blocks, while the f0 encoder consists of an embedding layer. The fused features are then fed into five levels of an acoustic encoder to extract audio features at different hierarchical levels: frame, word, phone, note, and sentence.

Each acoustic encoder begins with residual convolution blocks followed by a CMU Encoder for feature extraction. The encoder and decoder of the U-Net backbone include four downsampling and upsampling layers, with a downsampling rate of 16×. The U-Net module is further enhanced by two layers of Conformer blocks, where the Feed-Forward layers in each Conformer are replaced with FreqMOE layers. Each FreqMOE contains four experts, and the input features are equally split into four parts, each processed by an expert. The outputs of the four experts are then concatenated.

After feature extraction from the CMU Encoder, the sentence-level features are obtained by averaging across all frames and then they are expanded. For frame-level features, no further processing is performed. For the phone, word, and note levels, average pooling is applied based on boundary information. These pooled features then pass through convolutional layers followed by a Vector Quantization layer with a codebook size of 128 for feature filtering. Finally, the Length Regulator is applied to expand the features according to the corresponding boundary.

Hyperparameter		Model
Mel Encoder	Conv Kernel	3
	Conv Layers	2
	Hidden Size	256
Condition Encoder	Pitch Embedding	300
	UV Embedding	3
	Hidden Size	256
Vector Quantization	Code Num	128
	Hidden Size	256
U-Net	Kernel Size	3
	Enc & Dec Layers	4
	Downsampling Rate	16
	Hidden Size	256
FreqMOE Conformer	Kernel Size	9
	Head Num	4
	Layers	2
	Attention Hidden	256
	MOE Hidden	256
	Expert Num	4

Table 6: Hyperparameters of STARS.

The Global Style Predictor is composed of two layers of cross-attention, which predict the type of each class. The overall architecture parameters are shown in Table 6.

A.2 FreqMOE

Following previous audio generation systems with MOE (Zhang et al., 2025b,a), the FreqMOE (Frequency Mixture of Experts) module is designed to enhance the representation of input features by leveraging multiple experts, each processing a distinct subset of the input. The module operates by splitting the input feature map into multiple chunks and passing each chunk through a separate expert network. The outputs from all experts are then concatenated to form the final representation. Specifically, the FreqMOE can be expressed as:

$$\mathrm{FreqMOE}(\mathbf{X}) = \mathrm{Concat}_{k=1}^{K} E_k(\mathbf{X}^{(k)}), \quad (6)$$

where $\mathbf{X}_k \in \mathbb{R}^{T \times D/K}$ represents the k-th chunk of the input feature map, $\mathbf{X}$, split along its feature dimension. E_k denotes the k-th expert network, which processes $\mathbf{X}_k$.

B Audio-Phoneme Alignment

In this appendix, we briefly describe our alignment algorithm that synchronizes a sequence of

Setting	Metric	Technique Prediction									
		BUB	BRE	PHA	VIB	GLI	MIX	FAL	WEA	STR	**TEC**
Bilingual	F1	74.0	50.5	78.6	50.0	69.7	87.5	86.6	94.9	99.6	76.8
	ACC	98.5	89.0	94.6	96.3	84.2	81.1	91.1	90.7	97.3	91.4

Table 7: The objective results of phone-level technique prediction on the bilingual dataset.

setting	EMO	PAC	RNG	LAN	GEN
Single	52.5	71.3	59.0	-	100.0
Bilingual	48.6	71.8	76.9	100.0	100.0

Table 8: The results of the global style detection. EMO refers to emotion, PAC to pace, RNG to pitch range, LAN to language, and GEN to gender.

phoneme labels with the corresponding audio (or video) frames. The algorithm is based on dynamic programming and is inspired by Viterbi decoding, which efficiently finds the most likely alignment path through a state-space representing both phoneme and silence (or blank) predictions.

B.1 Overview

Given an audio signal, a neural network produces frame-level log-probabilities for phoneme classes as well as for silence. In addition, a boundary detection mechanism provides probabilities indicating the likelihood of transitions between phoneme segments. The alignment problem is then formulated as finding the optimal path that maximizes the overall likelihood, taking into account both the phoneme content and the temporal boundaries between phonemes.

B.2 Dynamic Programming Formulation

Let T be the number of time frames and L be the number of phonemes in the target sequence. We define a score matrix $\mathbf{D} \in \mathbb{R}^{T \times (2L+1)}$, where each column corresponds to a state representing either a phoneme or an interleaved blank state. The algorithm initializes $\mathbf{D}$ with scores computed from the first frame's predictions and then iteratively updates the matrix as follows:

$$D(t,k) = \max \begin{cases} D(t-1,k) + s_k^{(t)}, \\ D(t-1,k-1) + s_k^{(t)}, \\ D(t-1,k-2) + s_k^{(t)}, \end{cases}$$

Here, $s_k^{(t)}$ denotes the log-probability score at time t for state k, which is computed based on the phoneme, silence, and boundary predictions. A corresponding backtracking matrix $\mathbf{B}$ is maintained to record the optimal transition at each step.

B.3 Backtracking and Temporal Mapping

After processing all frames, the optimal alignment path is recovered by backtracking through $\mathbf{B}$ starting from the final frame. This path indicates, for each time frame, the most likely state (i.e., phoneme or blank) that generated the observation. Finally, by mapping the indices of phoneme states to time instants—using the known hop size of the audio frames—the algorithm produces onset and offset times for each phoneme. This procedure allows us to effectively synchronize phonetic labels with the audio signal.

B.4 Summary of the Algorithm

The overall alignment procedure can be summarized as follows:

1. **Initialization:** Set up the dynamic programming matrices $\mathbf{D}$ (for scores) and $\mathbf{B}$ (for backtracking) using the initial predictions.

2. **DP Matrix Update:** For each time frame $t = 1, \ldots, T-1$, update the scores for all states by considering self-transitions, transitions from the previous state, or skips (modeling boundaries).

3. **Backtracking:** Recover the optimal alignment path by backtracking through $\mathbf{B}$.

4. **Temporal Mapping:** Convert the alignment indices to temporal onset and offset times for each phoneme using the audio frame hop size.

C Details of Experiments

C.1 Dataset

The open-source singing dataset used in our experiments includes the Chinese and English subsets of GTSinger (Zhang et al., 2024c). We use the dataset under the CC BY-NC-SA 4.0 license. For the recordings, we select one male and one female professional singer, each paid $350 per hour, and they agree to make their contributions available for research purposes. During the recording sessions, the singers are instructed to apply and label the technique annotations at both the sentence and phoneme levels. Phoneme segmentation is refined using the Montreal Forced Aligner (MFA), with additional manual adjustments to ensure accuracy. The annotators are compensated at a rate of $15 per hour for their work.

C.2 Evaluation Metrics

For lyric alignment, we use two objective metrics: Boundary Error Rate (BER) and Intersection Over Union (IOU) score. The Boundary Error Rate (BER) measures the proportion of misplaced boundaries within 20 ms tolerance distance. The IOU score is defined as the ratio of the duration of the overlapping segment between two notes to the duration of the combined time span covered by both notes.

For note transcription, we use the objective metric COnPOff (Correct Onset, Pitch, and Offset). COnPOff assesses whether both the boundaries and pitch of the note are correctly predicted. Additionally, we compute the Raw Pitch Accuracy (RPA) to evaluate the overall pitch prediction performance. RPA is calculated by transforming both the ground truth (GT) and predicted note events into frame-level sequences and computing the matching scores.

In our evaluation process, we employ STARS to exclude silent notes and designate the boundaries enclosing each note as its onset and offset. For onset and offset evaluation, the tolerance is set to 50 ms, with the offset tolerance being the larger of 50 ms or 20% of the note duration. The pitch tolerance is set to 50 cents.

For the singing voice synthesis experiment, we randomly select 30 generated singing clips for subjective evaluation. Each generated sample is rated and evaluated by at least five professional listeners. In the MOS-Q evaluation, listeners focus on rating the quality and naturalness of the generated singing voice. In the MOS-C evaluation, they assess whether the generated singing matches the style of the given prompt. All scores are rated on a five-point scale. Each participant is paid $10 per hour.

C.3 Experiment

As shown in Table 7, we observe that when evaluating on a bilingual dataset, the overall prediction accuracy for the nine singing techniques is relatively high, comparable to the results obtained from a single-language dataset. This demonstrates the feasibility of our model for multilingual datasets.

According to the results presented in Table 8, we find that attributes such as language, gender, and vocal range, which are relatively fixed across the entire singing performance, yield better prediction results. n contrast, the model performs less effectively on attributes like emotion and vocal range, which may vary across different segments of the song. Further analysis of the dataset reveals that annotations for these attributes are typically provided at the sentence level, whereas emotion and vocal range fluctuate within different sections of a song. This variability leads to a decrease in prediction accuracy for individual segments of the singing voice.