AnchorAlign++: A Count-Aware and Anchor-Driven Neural Framework for Joint Lyric–Note Transcription and Robust Singing Assessment

Hu Wei, Gao Hongfeng
Beijing University of Chemical Technology
huwei@buct.edu.cn;2023210556@buct.edu.cn

***Abstract*—Automatic singing assessment relies on aligning sung notes with a reference score; however, achieving robust alignment remains challenging for beginners due to pitch instability and phrasing errors. We propose AnchorAlign++, an end-to-end framework that explicitly models the number of notes aligned to each lyric token as a structural variable and converts this discrete prediction into actionable temporal constraints through an anchor-driven inference strategy. The model consists of a shared acoustic encoder, a non-autoregressive note decoder for frame-level pitch and onset estimation, and an autoregressive lyric decoder that jointly predicts lyric tokens and their corresponding note counts. To further enhance robustness, we introduce a unidirectional acoustic–symbol fusion mechanism that injects frame-level pitch and onset cues into lyric decoding, thereby stabilizing long-tail note count prediction. We further propose a count–onset consistency loss, which couples discrete note counts with continuous onset probabilities to reduce boundary drift. During inference, the predicted note counts guide the selection of onset anchors, forming sharp lyric–note boundaries without manual correction. Experiments on multiple singing datasets demonstrate that AnchorAlign++ consistently improves transcription accuracy and alignment robustness, while maintaining practical computational efficiency for deployment, providing reliable automatic singing assessment for both novice and professional singers.**

***Index Terms*—Joint singing transcription, Automatic Singing Assessment, Lyric–Note Alignment, Anchor-driven Alignment**

I. Introduction

Automatic Singing Assessment aims [1], [2] to objectively evaluate a singer's intonation and rhythm by comparing a sung performance with a reference musical score. This task is important for applications such as music education and human–computer interaction.

However, producing a stable and reliable alignment between a singing performance and the score remains challenging in practice, especially for amateur or non-professional singers. In such scenarios, pitch drift, phrase breaks, and repeated fragments frequently occur, causing most existing methods that rely solely on continuous pitch trajectories to become highly unstable.

Most current systems adopt a relatively simple pipeline: note events transcribed by an Automatic Singing Transcription (AST) model [3]–[6] are aligned to the reference score using Dynamic Time Warping (DTW) [7], and the resulting alignment is then used for evaluation. Such approaches implicitly assume that the overall performance is largely correct, with only minor local errors, which is often unrealistic in beginner singing practice.

This work is supported in part by the Natural Science Foundation of Shandong Province under Grant ZR2024ZD19.

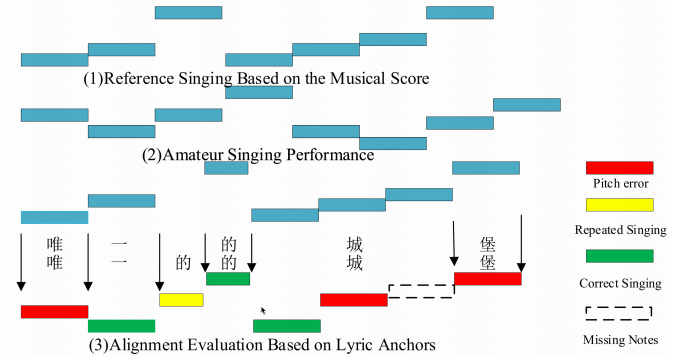

Fig. 1. Alignment Challenges in Real Singing Performances

A. Motivation and Hypothesis

As shown in Figure 1 our work is motivated by a practical observation: even when pitch accuracy is unstable, singers rarely sing incorrect lyrics. Compared to pitch trajectories, lyric sequences therefore provide a more stable and reliable structural signal for alignment.

A further challenge arises from *melisma*, a common singing phenomenon in which a single lyric unit is sung over multiple consecutive notes (whereas in typical singing, multiple lyric units rarely correspond to the same note).

Based on this observation, we propose the following hypothesis:

Explicitly modeling the number of notes associated with each lyric token, and using this structural prediction to constrain temporal alignment, can significantly improve robustness under pitch instability and melisma.

This hypothesis suggests that the note count per lyric token should not be treated as an implicit by-product of attention mechanisms or post-processing steps, but instead elevated to a first-class structural variable in the model. By doing so, the inherent one-to-many relationship between lyrics and notes can be directly represented and exploited during both training and inference.

B. Contributions

Based on the above motivation, we propose **AnchorAlign++**, a note-count-aware framework for joint lyric–note transcription and alignment. The main contributions of this work are summarized as follows:

- We introduce a note-count-aware joint transcription framework that explicitly predicts the number of notes aligned to each lyric token, enabling direct modeling of one-to-many lyric–note relationships.
- We design a unidirectional acoustic–symbol fusion mechanism that injects frame-level pitch and onset cues into the lyric decoding process, improving robustness for long-tailed note count prediction.
- We propose a note-count–onset consistency loss that couples discrete note counts with continuous onset probabilities, effectively mitigating boundary drift in lyric–note alignment.
- We develop an anchor-driven inference strategy that converts predicted note counts into high-confidence onset anchors, producing sharp lyric–note boundaries without manual correction.

II. Related Work

Automatic Singing Assessment [1] involves multiple closely related yet historically separated research tasks, including note-level singing transcription, lyric transcription, alignment between lyrics and notes (or musical scores), and the final computation of similarity measures or quality judgments for evaluation purposes. According to research objectives and technical paradigms, existing work can be broadly categorized into the following directions.

A. Traditional Singing Assessment

The most traditional paradigm for automatic singing assessment follows a **"transcribe–align–score"** pipeline. In this framework, pitch trajectories or note sequences are first extracted from the singing audio, then dynamically aligned to a reference score (or a reference performance), and finally evaluated using pitch- and rhythm-related metrics computed from the alignment results.

Among these methods, **Dynamic Time Warping (DTW)** and its variants are the most widely adopted alignment tools, commonly applied to continuous pitch contours, chroma features, or symbolic event sequences. In some systems, the aligned results are further fed into manually designed scoring functions or used to estimate correlations with expert ratings. The main advantages of this paradigm lie in its clear processing flow, strong interpretability, and natural compatibility with pedagogical singing assessment criteria.

However, this approach heavily depends on the stability of upstream transcription results. When singing performances exhibit pitch drift, phrasing breaks, or repeated fragments, alignment based purely on pitch trajectories often suffers from global drift. These limitations significantly reduce the robustness of traditional assessment systems in real-world singing scenarios.

B. Automatic Singing Transcription

Automatic Singing Transcription [3]–[6] aims to convert singing audio into a sequence of discrete note events, where each note is typically characterized by its pitch and onset (and sometimes offset) time. This task has a long research history in the Music Information Retrieval community and is supported by relatively mature evaluation protocols and benchmark datasets.

Classical approaches include systems based on Hidden Markov Models and rule-based post-processing pipelines. Modern deep learning methods commonly adopt convolutional or Transformer-based architectures to predict **frame-level pitch classes and onset probabilities**, often trained using combinations of cross-entropy and CTC losses [8]. Nevertheless, these methods generally assume relatively stable onset boundaries. Under conditions such as pitch glides, vibrato, or expressive ornamentation, their performance degrades, which in turn negatively impacts downstream assessment tasks.

C. Automatic Lyric Transcription

Automatic Lyrics Transcription [9], [10] is commonly regarded as a speech-recognition-like task in the singing domain, aiming to automatically recognize textual content from sung vocals. Compared to conventional automatic speech recognition, ALT is more challenging due to the presence of prolonged vowels, large pitch variations, and strong interference from musical accompaniment.

With the development of deep learning techniques, end-to-end modeling frameworks, such as CTC-based acoustic models and encoder–decoder architectures, have been widely adopted to better capture the temporal structure and long-range dependencies in singing voice. In recent years, transfer learning based on self-supervised speech representations has become prevalent in ALT research, enabling significant performance gains by adapting pretrained models to singing corpora under limited data conditions [11].

To address issues such as severe accompaniment interference and difficulties in aligning lyrics with melody, some studies incorporate vocal separation as a preprocessing step or employ multi-task learning to jointly model musical attributes such as pitch and rhythm. In addition, integrating language models and lyric prior information has been shown to further improve the stability and accuracy of transcription. Despite these advances, automatic lyrics transcription remains highly challenging in scenarios involving complex accompaniment, diverse singing styles, and cross-lingual settings.

D. Joint Lyric–Note Modeling and Unified Transcription Frameworks

Recent studies have explored unified transcription frameworks that jointly model lyrics and musical notes within a single system. These approaches typically adopt **multi-branch neural architectures**, where one branch focuses on modeling acoustic and melodic information, while another branch captures linguistic content. However, most existing

methods **lack explicit modeling and constraints on the alignment** between lyrics and notes.

Representative systems such as **SongTrans** [12] propose a unified model for joint lyrics and note transcription using a relatively simple end-to-end design. Nevertheless, due to the **highly imbalanced distribution of melismatic singing** and the resulting long-tail phenomenon in real-world datasets, the performance of such methods is often limited in complex scenarios. **STARS** [13] further introduces a structured joint modeling strategy, learning shared representations for lyric and note sequences, which improves overall transcription consistency to some extent.

Overall, these joint modeling approaches **reduce the reliance on manual post-processing** and facilitate large-scale automatic annotation of singing data. However, **most existing unified models still rely on implicit alignment mechanisms**, which tend to favor dominant one-to-one mappings during training. When confronted with long-tail melismatic distributions, such implicit biases often lead to **boundary drift**, thereby degrading the **stability and interpretability** of alignment results in both evaluation and practical applications.

III. Methodology

A. Data Augmentation

To improve robustness, we apply three data augmentation strategies: **noise mixing**, **volume adjustment**, and **temporal label smoothing**.

Noise Mixing. We mix clean waveforms with the MUSAN noise dataset [3], where the signal-to-noise ratio (SNR) is randomly sampled from $[8, 20]$ dB.

Volume Adjustment. We apply a random global gain to each waveform to simulate different recording conditions and loudness variations.

Temporal Label Smoothing. Since precise note boundaries in singing are inherently ambiguous, we follow prior work [4], [14] and adopt a label smoothing strategy. By applying a short convolutional smoothing window to the target labels along the time axis, we reduce the model's sensitivity to exact boundary positions and improve training stability.

B. Architecture Overview

As illustrated in Figure 2,AnchorAlign++ consists of three components:

1) **Shared Acoustic Encoder** producing frame-level features.
2) **Note Decoder** predicting frame-level pitch and onset probabilities.
3) **Lyric–Count Decoder** jointly predicting lyric tokens and per-token note counts, conditioned on fused acoustic cues.

The design follows a locality principle: note events are mainly determined by local acoustics (parallel prediction), while lyric and count prediction depend on longer-range linguistic context (autoregressive decoding).

C. Acoustic Encoder

We adopt Wav2Vec2 (XLS-R-1B) [15], [16] as a shared encoder, followed by a linear projection to $d_{\text{model}} = 512$. The downsampling is performed at 50 Hz, and we use this to compute the exact frame length for each audio input. This allows us to construct a boolean padding mask, preventing irrelevant frames from interfering with attention during large-batch training. Given an input waveform x, the encoder outputs a feature sequence $\mathbf{H} \in \mathbb{R}^{T\times d}$ and a mask $\mathbf{M}$, which are consumed by the downstream decoders.

D. Note Decoder

Given the encoder feature sequence $\mathbf{H}$, the Note Decoder applies a Transformer stack to predict frame-level pitch and onset logits:

$$\hat{\mathbf{P}} \in \mathbb{R}^{T\times C_{\text{pitch}}}, \quad \hat{\mathbf{O}} \in \mathbb{R}^{T}. \tag{1}$$

It also outputs pitch-side hidden states $\mathbf{H}_{\text{pitch}}$ for cross-modal conditioning in the Lyric Decoder.

Our Note Decoder adopts a non-autoregressive design: frame-wise pitch and onset mainly rely on short-term acoustics, so parallel prediction is natural, reduces latency, and avoids left-to-right error accumulation.

E. Acoustic–Symbol Fusion

In our task, note count prediction is a core bottleneck due to three factors: (1) **High uncertainty** — for the same lyric syllable, the number of aligned notes can range from 1 to 14, making it difficult for the AR branch to capture such variations using standard CE loss. Moreover, the long-tail distribution of real-world lyric-note alignments, where one-to-many mappings is less frequent but still significant, further complicates the task. The uneven distribution of these mappings in the training data makes it harder to train a robust model. (2) **Implicit acoustic cues** — note counts depend more on melody structure, which is not explicitly encoded in the lyric modality. (3) **Error propagation** — inaccurate counts in an end-to-end system cause lyric boundary shifts, leading to "whole-line drift" alignment errors, as seen in SongTrans [12].

To address these issues, we design a **unidirectional acoustic–Symbol Fusion** that injects note cue produced by the Note Decoder into the Lyric Decoder. This design explicitly exploits frame-level pitch and onset information as structural constraints on lyric–note mapping. **This fusion is essential because note counts strongly depend on melody structure, which cannot be reliably inferred from lyrics alone.**During fusion, we enhance the Lyric Decoder's acoustic features with both pitch embeddings and onset embeddings:

$$\mathbf{H}_{\text{fuse}} = \mathbf{H} + \text{Embed}\big(\arg\max \hat{\mathbf{P}}\big) + \sigma(\hat{\mathbf{O}}) \cdot \mathbf{v}_{\text{onset}}, \tag{2}$$

where $\text{Embed}(\cdot)$ maps the most likely pitch class to an embedding, $\sigma(\cdot)$ is the sigmoid, and $\mathbf{v}_{\text{onset}}$ is a learnable vector. In addition, the pitch-side hidden states $\mathbf{H}_{\text{pitch}}$ are provided as auxiliary memory. The Lyric Decoder consumes $\mathbf{H}_{\text{fuse}}$ together with $\mathbf{H}_{\text{pitch}}$.

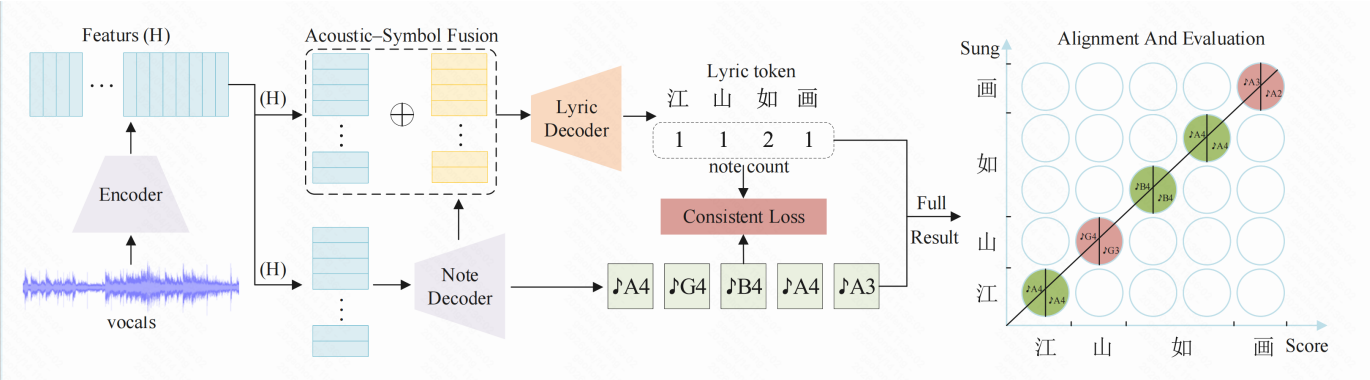

Fig. 2. The overall framework of AnchorAlign++.

F. Lyric–Count Decoder

The Lyric–Count Decoder produces two synchronized outputs: a **pinyin-level lyric token sequence** $\mathbf{y}_{1:N}$ and the corresponding **per-token note counts** $\mathbf{c}_{1:N}$, where each c_i denotes the number of musical notes aligned to the pinyin token y_i. Instead of predicting Chinese characters, the decoder operates on a **predefined vocabulary of approximately 406 pinyin units**, which provides a compact and pronunciation-oriented representation that is more suitable for sequence modeling and temporal alignment.

The decoder is implemented using stacked self-attention and cross-attention layers, and attends to two memory sources: the fused acoustic–symbol representations $\mathbf{H}_{\text{fuse}}$ and the pitch-side hidden states $\mathbf{H}_{\text{pitch}}$. This design allows melodic and onset-related acoustic cues to be explicitly injected into the autoregressive decoding process, while preserving long-range linguistic dependencies in the pinyin sequence.

During training, the decoder takes the prefix input $[\text{BOS}], \mathbf{y}_{1:L-1}$ and jointly predicts the next pinyin token y_t together with its associated note count c_t. To further stabilize the alignment between symbolic tokens and acoustic frames, we introduce an **auxiliary CTC head** on top of $\mathbf{H}_{\text{fuse}}$. This hybrid AR–CTC formulation is consistent with recent lyric transcription systems based on large-scale sequence models, and helps reduce boundary ambiguity during training.

The Lyric–Count Decoder operates in an **autoregressive manner**: both the pinyin token sequence and the per-token note counts are conditioned on left-to-right linguistic context and evolving alignment information. By explicitly modeling note counts as a first-class structural variable, the decoder enables robust learning of one-to-many lyric–note relationships under pitch instability and melismatic singing.

G. Total Loss and Training

Our training objective consolidates frame-level supervision from the **Note Decoder**, sequence-level supervision from the **Lyric Decoder**, and an additional **count–onset consistency term** that explicitly couples note counts with onset probabilities:

$$\mathcal{L} = \mathcal{L}_{\text{note}} + \mathcal{L}_{\text{lyric}} + \lambda_{\text{cons}}\,\mathcal{L}_{\text{cons}}. \tag{3}$$

The Note Decoder predicts pitch classes and onset probabilities at each frame:

$$\mathcal{L}_{\text{note}} = \lambda_{\text{p-ce}}\,\mathcal{L}_{\text{p-ce}} + \lambda_{\text{p-ctc}}\,\mathcal{L}_{\text{p-ctc}} + \lambda_{\text{on}}\,\mathcal{L}_{\text{onset}}. \tag{4}$$

Here, $\mathcal{L}_{\text{p-ce}}$ is a cross-entropy over pitch classes, $\mathcal{L}_{\text{p-ctc}}$ is a CTC term encouraging monotonic frame-to-pitch alignment, and $\mathcal{L}_{\text{onset}}$ is a focal-weighted BCE for sparse onsets [4].

The Lyric Decoder jointly predicts lyric tokens and per-token note counts:

$$\mathcal{L}_{\text{lyric}} = (1-\lambda_{\text{ctc}})\Big(\mathcal{L}_{\text{char-ce}} + \lambda_{\text{cnt}}\,\mathcal{L}_{\text{cnt-ce}}\Big) + \lambda_{\text{ctc}}\,\mathcal{L}_{\text{ctc}}, \tag{5}$$

where $\mathcal{L}_{\text{char-ce}}$ is token-level cross-entropy (ignoring <AP> / <SP>), $\mathcal{L}_{\text{cnt-ce}}$ is a class-weighted cross-entropy mitigating the heavy-tailed count distribution, and $\mathcal{L}_{\text{ctc}}$ is an auxiliary CTC applied on $\mathbf{H}_{\text{fuse}}$.

Count–Onset Consistency term.: For the j-th lyric token, we define its temporal span from $\text{prev}(j)$ to $\text{end}(j)$ and require that the integral of onset probabilities matches its predicted note count c_j:

$$\mathcal{L}_{\text{cons}} = \frac{1}{J}\sum_{j=1}^{J}\left(\sum_{t=\text{prev}(j)}^{\text{end}(j)}\sigma(\hat{o}_t) - c_j\right)^2. \tag{6}$$

This term directly couples a discrete event count (note count) with a continuous intensity (onset probability), constraining the alignment to reduce boundary drift and improving segmentation accuracy under long-tail melisma.

We set $\lambda_{\text{p-ce}} = 1.0$, $\lambda_{\text{p-ctc}} = 0.3$, $\lambda_{\text{on}} = 10.0$, $\lambda_{\text{cnt}} = 1.0$, $\lambda_{\text{ctc}} = 0.5$, and $\lambda_{\text{cons}} = 3.0$. Experiments show that performance is not sensitive to these weights, so we use this fixed configuration throughout.

H. Inference

The goal of inference is to determine, *within the sung performance itself*, the correspondence between lyric tokens and musical notes, i.e., to assign a set of note boundaries to each decoded lyric token. The Lyric–Count Decoder first outputs a lyric token sequence $\mathbf{y}_{1:N}$ together with the corresponding per-token note counts $\mathbf{c}_{1:N}$, where c_i denotes the number of notes aligned to lyric token y_i. Meanwhile, the Note

Decoder produces a frame-level onset probability sequence $\hat{\mathbf{o}} = \{\hat{o}_t\}_{t=1}^{T}$.

We convert the discrete note count predictions into explicit temporal structure by anchor selection. Specifically, the expected total number of note events in the sung performance is estimated as

$$K = \sum_{i=1}^{N} c_i.$$

We then select the K frames with the highest onset probabilities from $\hat{\mathbf{o}}$ as note onset anchors, while enforcing a minimum temporal separation to avoid redundant detections. This results in a temporally ordered anchor set $\mathcal{A} = \{a_1 < a_2 < \cdots < a_K\}$.

Based on the predicted note counts $\mathbf{c}_{1:N}$, the anchors are grouped to form a deterministic lyric–note correspondence. For the i-th lyric token, we define its covered note index range as

$$B(i) = 1 + \sum_{m=1}^{i-1} c_m, \qquad E(i) = \sum_{m=1}^{i} c_m.$$

Accordingly, lyric token y_i is aligned to the note set $\{\hat{n}_{B(i)}, \ldots, \hat{n}_{E(i)}\}$, where each predicted note $\hat{n}_k$ is associated with onset time a_k and the corresponding predicted pitch $\hat{p}_k$. This procedure establishes a monotonic and deterministic one-to-many lyric–note mapping without relying on attention weights or global DTW paths.

I. Evaluation

Evaluation is performed by using lyrics as an intermediate alignment layer to map sung notes to reference score notes and then comparing their pitch attributes. Let $\mathbf{s}_{1:M}$ denote the lyric token sequence of the reference score, and let $\mathcal{S} = \{s_1, \ldots, s_J\}$ be the corresponding score note sequence. For each score lyric token s_j, the covered score note index range $[B_s(j), E_s(j)]$ is given by the score annotation.

We first align the sung lyric sequence $\mathbf{y}_{1:N}$ to the score lyric sequence $\mathbf{s}_{1:M}$ using Dynamic Time Warping (DTW), resulting in a monotonic alignment path

$$\mathcal{P} = \{(i_\ell, j_\ell)\}_{\ell=1}^{L}, \qquad i_\ell \in [1, N],\ j_\ell \in [1, M],$$

which minimizes the cumulative lyric-level matching cost. From this path, we derive a mapping from sung lyric tokens to score lyric tokens, denoted by

$$g(i) = \mathrm{repr}\left(\{\, j \mid (i, j) \in \mathcal{P} \,\}\right),$$

where $\mathrm{repr}(\cdot)$ selects a representative score index (e.g., the index with minimum accumulated cost).

Using lyrics as an intermediate representation, sung notes are then mapped to score notes. For a predicted note $\hat{n}_k$, we first identify the associated sung lyric token

$$i(k) = \min\{\, i \mid B(i) \le k \le E(i) \,\}.$$

The corresponding score lyric token is given by $j = g(i(k))$, and $\hat{n}_k$ is mapped to the score note range $[B_s(j), E_s(j)]$. A concrete score note index is then assigned by preserving the relative order within the lyric segment:

$$\pi(k) = B_s(j) + \big(k - B(i(k))\big),$$

with boundary truncation applied if $\pi(k)$ exceeds $E_s(j)$.

Finally, pitch evaluation is conducted for each matched note pair $(\hat{n}_k, s_{\pi(k)})$ by computing the pitch deviation

$$\Delta p_k = \mathrm{PitchDiff}\big(\hat{p}_k, p_{\pi(k)}\big),$$

where $p_{\pi(k)}$ denotes the reference pitch of the score note. The overall singing assessment result is obtained by aggregating the pitch comparison outcomes over all matched notes.

IV. Experiments and Evaluation

A. Datasets

We evaluate the proposed method on multiple datasets to assess its generalization capability:

- **GTSinger** [17]: a multilingual singing dataset containing more than **80 hours** of high-quality recordings, with corresponding musical score information collected for each song. In this work, we mainly use the Chinese songs from this dataset;
- **M4Singer** [18]: a Mandarin singing corpus of approximately **26.5 hours** after post-processing, covering a wide range of musical styles;
- **ISMIR-2014** [19] and **MIR-ST500** [15]: standard benchmark datasets for AST evaluation.

All datasets are split at the **song level** into **80% training** and **20% testing** sets. All audio recordings are resampled to **16 kHz**, and frame-level features are extracted at a frame rate of **50 Hz**.

We conduct a comprehensive evaluation of the proposed method, focusing on three aspects: (1) note transcription performance, (2) lyric transcription performance, and (3) overall system performance for automatic singing assessment.

B. Training Configuration

We train all models using the **AdamW** optimizer with a **cosine learning rate decay** schedule to improve convergence stability. The initial learning rate is set to 1×10^{-4} and is gradually decayed to 1×10^{-6} over the course of training. The AdamW momentum parameters are set to $\beta_1 = 0.9$ and $\beta_2 = 0.98$, with a weight decay coefficient of **0.01**.

Training is conducted in a distributed manner on **eight NVIDIA RTX 4080-Super GPUs**. The per-GPU batch size is set to **4**, resulting in a global batch size of **32**. All input audio sequences are padded to a fixed length to ensure efficient batch-wise computation. We adopt **mixed-precision training** to reduce memory usage and accelerate training.

The total number of training epochs is set to **30**. During the early stage of training, we apply a **linear warm-up** over the first **10,000 steps** to mitigate unstable gradients when training large-scale models.

TABLE I
PERFORMANCE COMPARISON ON MIR-ST500 AND ISMIR-2014 DATASETS

Method	MIR-ST500		ISMIR-2014	
	COnP	COn	COnP	COn
Tony	0.4078	0.5345	0.6170	0.6763
ROSVOT	0.6590	0.7210	0.6833	0.8128
CE+CTC	0.7437	0.7962	0.7673	0.9301
Qiu & Zhang	0.7758	**0.8161**	0.7728	0.9376
Ours	**0.7861**	0.8112	**0.7752**	**0.9403**

TABLE II
LYRICS TRANSCRIPTION AND EVALUATION RESULTS.

Method	CER (%)	NCP (%)	NMA (%)	SEA (%)
ROSVOT+DTW	—	—	78.33	77.42
STARS	10.12	—	89.71	84.36
SongTrans	9.61	92.17	88.36	85.31
Ours	**9.33**	**95.22**	**90.17**	**87.59**
Ours (–ASF)	9.57	90.89	86.22	86.03
Ours (–CL)	9.61	93.01	88.43	86.49
Ours (–PN/PM)	—	—	88.75	88.67

TABLE III
AVERAGE INFERENCE LATENCY FOR 10-SECOND AUDIO INPUT

Method	STARS	SongTrans	Proposed
Latency (s)	1.37	1.42	0.67

C. Note Transcription Results

We evaluate note transcription performance using two standard metrics:

- **COn (Correct Onsets)**: the proportion of notes with correctly detected onset times;
- **COnP (Correct Onsets and Pitches)**: the proportion of notes whose onset times and pitches are both correctly predicted [19].

All results are reported in terms of **F1 score**, and silent notes are excluded from evaluation. An onset prediction is considered correct if the temporal deviation from the reference onset is within $\pm$**50 ms**. A pitch prediction is considered correct if the pitch deviation is within **50 cents**.

We compared our method with several baselines, including *Tony* (HMM-based) [20], *ROSVOT* [4], *CE+CTC* [5], and the current SOTA by *Qiu & Zhang* [21]. Results are shown in Table I. Our approach achieves the best or near-best performance on almost all cases across both datasets, demonstrating its effectiveness for note transcription.

D. Lyric Transcription Results

In this work, lyric transcription experiments are conducted exclusively on **Chinese songs**. We therefore adopt **Character Error Rate (CER)** as the evaluation metric.

When computing evaluation scores, padding tokens such as `<AP>` and `<SP>` are excluded.

We compared against three baselines, *ROSVOT+DTW [4],STARS [13]* and the current SOTA *SongTrans* [12]. Results are shown in Table II. Our method achieves the best results on both metrics, demonstrating strong performance in lyric transcription and note count prediction.

E. Overall System Performance

For end-to-end transcription and alignment, we evaluate the overall system performance using two complementary metrics:

- **NMA (Note–Score Matching Accuracy):** A reference note is considered correctly matched if and only if all of the following conditions are satisfied: (i) the transcribed note is aligned to the correct score position after the alignment procedure; (ii) the onset deviation from the reference onset is within ± 50 ms. NMA is defined as the proportion of reference notes that fulfill the above criteria.
- **SEA (Singing Evaluation Accuracy):** SEA measures the proportion of notes for which the system provides a correct singing assessment decision. In particular, a note is counted as correct only when it is not only aligned to the correct score position, but also transcribed accurately in terms of its musical attributes (e.g., pitch). This metric explicitly avoids the case where the alignment is correct but transcription errors lead to incorrect evaluation feedback.

The results are summarized in Tables II and III. Our method achieves the best or near-best performance on both metrics, demonstrating the effectiveness of the proposed *anchor-driven acoustic–symbol alignment mechanism*.

Beyond accuracy, AnchorAlign++ also exhibits a significant efficiency advantage over baseline systems such as SongTrans. Specifically, the proposed model reduces the number of parameters by approximately **52%**, while decreasing the average inference latency for a **10-second audio segment** from **1.42 s** to **0.67 s**. These improvements make AnchorAlign++ particularly suitable for **real-time** and **resource-constrained** singing assessment applications.

F. Ablation Study

We conduct ablation experiments to analyze the contribution of each component, including:

- **–ASF**: removing the acoustic–symbol fusion module;
- **–CL**: removing the note count–onset consistency loss;
- **–PN/DP**: removing phonetic normalization and dynamic programming refinement.

The results are summarized in TableII. Removing the acoustic–symbol fusion module leads to a significant performance drop, particularly in NCP (**–4.3%**) and NMA (**–4.0%**), indicating that interaction between acoustic and lyric features is critical for accurate alignment.

When the consistency loss is removed, overall performance remains relatively strong, but consistent degradations are observed in NCP (**–2.2%**) and SEA (**–1.1%**), suggesting that enforcing consistency between discrete note counts and onset probabilities stabilizes alignment and evaluation.

Finally, removing phonetic-level alignment results in degradation in both NMA (**–1.4%**) and SEA (**–0.9%**), highlighting

the importance of pronunciation-level cues for precise matching and assessment.

Overall, these results confirm that all components of the proposed system are indispensable for achieving optimal performance.

V. Conclusion

We propose **AnchorAlign++**, a **count-aware and anchor-driven framework** for joint lyric–note transcription and robust singing assessment.

By explicitly modeling per-token note counts, injecting acoustic cues into lyric decoding, coupling discrete note counts with onset probabilities through a consistency loss, and converting predicted counts into anchor-based constraints during inference, the proposed method significantly improves alignment stability under conditions such as **pitch drift** and **long-tail melisma**. At the same time, it reduces the **model size and inference latency by more than 50%**, making it well suited for **real-time** and **large-scale deployment** scenarios.

In addition, we introduce **phonetic normalization** to improve alignment and evaluation accuracy at the pronunciation level. Although the experiments in this work are conducted only on **Mandarin datasets**, the framework itself is **language-agnostic** by design and can be readily extended to other languages with minimal modification. For example, Mandarin pinyin mappings can be replaced with phoneme-based representations for English, making AnchorAlign++ a promising solution for **multilingual singing assessment**.

References

[1] W. Yang, X. Wang, B. Tian, W. Xu, and W. Cheng, "A multi-stage automatic evaluation system for sight-singing," *IEEE Transactions on Multimedia*, vol. 25, pp. 3881–3893, 2022.

[2] J. Huang, Y.-N. Hung, A. Pati, S. K. Gururani, and A. Lerch, "Score-informed networks for music performance assessment," *arXiv preprint arXiv:2008.00203*, 2020.

[3] D. Snyder, G. Chen, and D. Povey, "Musan: A music, speech, and noise corpus," *arXiv preprint arXiv:1510.08484*, 2015.

[4] R. Li, Y. Zhang, Y. Wang, Z. Hong, R. Huang, and Z. Zhao, "Robust singing voice transcription serves synthesis," *arXiv preprint arXiv:2405.09940*, 2024.

[5] J.-Y. Wang and J.-S. R. Jang, "Training a singing transcription model using connectionist temporal classification loss and cross-entropy loss," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 31, pp. 383–396, 2022.

[6] Z.-S. Fu, L. Su *et al.*, "Hierarchical classification networks for singing voice segmentation and transcription." in *ISMIR*, 2019, pp. 900–907.

[7] M. Brown and L. Rabiner, "Dynamic time warping for isolated word recognition based on ordered graph searching techniques," in *ICASSP '82. IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 7, 1982, pp. 1255–1258.

[8] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of the 23rd international conference on Machine learning*, 2006, pp. 369–376.

[9] L. Ou, X. Gu, and Y. Wang, "Transfer learning of wav2vec 2.0 for automatic lyric transcription," *arXiv preprint arXiv:2207.09747*, 2022.

[10] J. Syed, I. M. Higgs, O. Cífka, and M. Sandler, "Exploiting music source separation for automatic lyrics transcription with whisper," *arXiv preprint arXiv:2506.15514*, 2025.

[11] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," in *International conference on machine learning*. PMLR, 2023, pp. 28 492–28 518.

[12] S. Wu, J. He, R. Yuan, H. Wei, X. Wei, C. Lin, J. Xu, and J. Lin, "Songtrans: An unified song transcription and alignment method for lyrics and notes," *arXiv preprint arXiv:2409.14619*, 2024.

[13] W. Guo, Y. Zhang, C. Pan, Z. Zhu, R. Li, Z. Chen, W. Xu, F. Wu, and Z. Zhao, "Stars: A unified framework for singing transcription, alignment, and refined style annotation," *arXiv preprint arXiv:2507.06670*, 2025.

[14] Y. Zhang, R. Huang, R. Li, J. He, Y. Xia, F. Chen, X. Duan, B. Huai, and Z. Zhao, "Stylesinger: Style transfer for out-of-domain singing voice synthesis," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 17, 2024, pp. 19 597–19 605.

[15] A. Babu, C. Wang, A. Tjandra, K. Lakhotia, Q. Xu, N. Goyal, W.-N. Hsu, M. Auli, J. Pino, and A. Baevski, "XLS-R: Self-supervised cross-lingual speech representation learning at scale," in *Interspeech*, 2021.

[16] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," *Advances in neural information processing systems*, vol. 33, pp. 12 449–12 460, 2020.

[17] Y. Zhang, C. Pan, W. Guo, R. Li, Z. Zhu, J. Wang, W. Xu, J. Lu, Z. Hong, C. Wang *et al.*, "Gtsinger: A global multi-technique singing corpus with realistic music scores for all singing tasks," *Advances in Neural Information Processing Systems*, vol. 37, pp. 1117–1140, 2024.

[18] L. Zhang, R. Li, S. Wang, L. Deng, J. Liu, Y. Ren, J. He, R. Huang, J. Zhu, X. Chen *et al.*, "M4singer: A multi-style, multi-singer and musical score provided mandarin singing corpus," *Advances in Neural Information Processing Systems*, vol. 35, pp. 6914–6926, 2022.

[19] E. Molina, A. M. Barbancho-Perez, L. J. Tardon-Garcia, I. Barbancho-Perez *et al.*, "Evaluation framework for automatic singing transcription," 2014.

[20] M. Mauch, C. Cannam, R. Bittner, G. Fazekas, J. Salamon, J. Dai, J. Bello, and S. Dixon, "Computer-aided melody note transcription using the tony software: Accuracy and efficiency," 2015.

[21] Y. Qiu, J. Zhang, Y. Shan, and J. Zhou, "Enhancing note-level singing transcription model with unlabeled and weakly labeled data," in *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2024, pp. 341–345.