# Exploring Masked CE Losses to Enhance Word Offset Estimation in CTC-based Lyrics-to-Audio Alignment

**Tian Cheng, Tomayasu Nakano, and Masataka Goto**
National Institute of Advanced Industrial Science and Technology (AIST), Tsukuba, Japan
tian.cheng@aist.go.jp t.nakano@aist.go.jp m.goto@aist.go.jp

## ABSTRACT

*Lyrics-to-audio alignment is an important task for real-world applications such as karaoke systems. Despite alignment performance improved with the release of large datasets and the utility of advanced deep learning models, accurate word offset estimation remains challenging. To address this problem, we extend our previously proposed masked cross-entropy (CE) loss by proposing new masks to enforce model predictions at masked frames with frame-wise phoneme labels derived from word-level annotations. We train a Convolutional Recurrent Neural Network (CRNN) by using both the masked CE loss and the Connectionist Temporal Classification (CTC) loss. By comparing the results obtained by using different masks in the masked CE loss, we find that word offset estimation performance is improved by using masks which cover all silent frames. In addition, we find that masks on word onset frames are essential for improving word onset estimation performance. We achieve comparable word onset estimation results and provide benchmark word offset estimation results for future research.*

## 1. INTRODUCTION

In this paper, we address the lyrics-to-audio alignment task, which aims to synchronize lyrics with musical audio. There are two inputs for a lyrics-to-audio alignment system: the musical audio (a mix of singing and musical accompaniment) and the corresponding lyrics. The output is time-synchronized lyrics, with the start time (onset) and end time (offset) of each lyric unit [1]. Depending on the application scenario, the time-synchronized lyrics can be categorized into different granularity levels: such as line, word, and phoneme levels, as illustrated in Figure 1. This task is crucial for real-world applications that require accurate, large-scale time-synchronized lyrics, such as karaoke systems, lyrics-based music retrieval [2], and lyrics animation for music videos [3].

Lyrics-to-audio alignment is a challenging task due to the unique characteristics of singing voice and the interference introduced by musical accompaniment. Singing is generally more expressive than speech [2, 4, 5], with lyrics sung with varying dynamics, pitches, and durations. This variability in pronunciation makes it difficult to build an effective acoustic model. Furthermore, the musical accompaniment is highly correlated with the singing voice, which reduces the effective signal-to-noise ratio for singing voice processing. To address these challenges, some methods [6, 7, 8, 9, 10] separated singing vocals from the mixed audio by using advanced source separation methods [11] before alignment. Alternatively, other methods exploited temporal correlations between lyrics and musical elements, such as melody, chords and accompaniment. They improved lyrics-to-audio alignment performance by learning it jointly with other related tasks, such as chord estimation [12], singing pitch detection [9], singing voice separation [13], and singing voice synthesis [14].

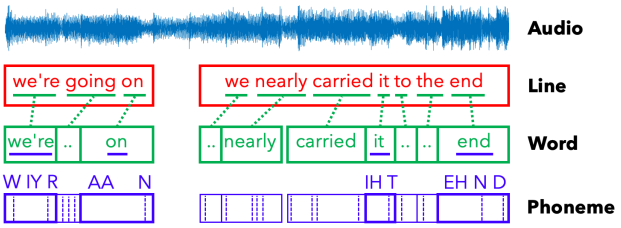
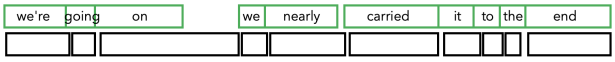


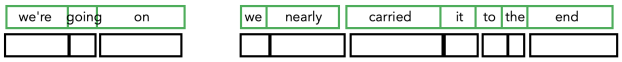

**Figure 1**: An illustration of the lyrics-to-audio alignment task with the input audio in waveform and the output time-synchronized lyrics in three levels: line, word, and phoneme levels.

With the release of the DALI dataset (a large Dataset of synchronised Audio, LyrIcs and notes) [15, 16], lyrics-to-audio alignment performance has been improved substantially [17, 9, 18, 19]. However, most available datasets provide only weak (line-level or word-level) annotations, lacking frame-wise (phoneme-level) annotations necessary for supervised training. To train models using such weak annotations, researchers have developed different approaches. Gupta et al. [20] employed a conventional automatic speech recognition method, using predicted frame-wise phoneme labels as ground truth for training. Stoller et al. [21] proposed an end-to-end model based on the Wave-U-Net architecture and trained it using the Connectionist Temporal Classification (CTC) loss [22]. Other methods leveraged cross-modal models by mapping audio and text embeddings into a shared latent space via contrastive learning [23, 19]. Kang et al. [18] acheived lyrics-to-audio alignment by reformulating the alignment task as a word onset estimation task and trained the model with the cross-entropy (CE) loss.

Among these methods, the CTC loss has attracted increasing attention in recent work [7, 24, 9, 10], because it simplifies model training by enabling learning from weak,




(a) Estimated results by using the method in [25] (with mask_p)




(b) Estimated results by using the proposed method (with onsets obtained by using mask_onset and offsets obtained by using mask_onset_silence)

**Figure 2**: Estimated word onsets and offsets by using the method in [25] and the proposed method, with the ground truth indicated by the above green boxes. See Section 3.2 for details of the masks.

line-level annotations without explicit alignment. However, the CTC loss is suitable for transcription tasks but not sensitive to the precise timing of individual elements in a sequence, which may limit alignment accuracy. To address this limitation, Teytaut et al. [24] added a spectral reconstruction loss to the CTC loss to improve the temporal coherence in the model's predictions. Huang et al. [9] proposed a joint learning approach that integrates pitch estimation to enhance alignment accuracy.

In our previous work [25], we also applied a CTC-based model and proposed a masked CE loss to mitigate the timing imprecision inherent in the CTC loss. In the masked CE loss, frame-wise phoneme labels were derived from word-level annotations at key frames (word onset and offset frames, as well as silent frames), with details shown in Section 3.1. Then, we enforced the model phoneme predictions at the key frames by using the masked CE loss with a mask on these frames. The experimental results showed that incorporating the masked CE loss improved word onset estimation. However, we observed incorrect offsets in the results: a word often ends just before the next word with the silent frames undetected, as shown in the Figure 2(a).

Therefore, in this paper, we work on improving word offset estimation performance in the lyrics-to-audio alignment system. We extend our previously proposed masked CE loss [25] by proposing new masks to explore the effects of different masks on alignment. We achieve better results on both word onset and offset estimation by using the new masks. By carefully examining the results of using different masks in the masked CE loss, we find that using masks covering all silent frames enhances word offset estimation; and masks on word onsets are essential for improving on word onset estimation. We combine the best onset and offset estimations as the output of the proposed method, as shown in Figure 2(b). The contributions of this paper are summarized as follows:

- We propose four new masks for the masked CE loss, improving word onset and offset estimation performance.
- We compare the effects of using different masks to find the essential frames in the masks for performance improvements on word onset and offset estimation, respectively.
- We conduct evaluation on word offsets. Since there are no existing results on word offset estimation, we present our results as a benchmark for future research.

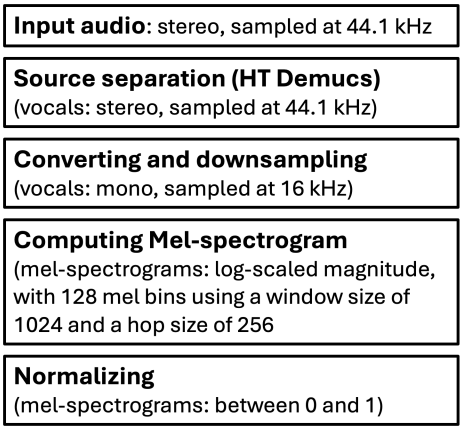



**Figure 3**: Audio pre-processing

The rest of this paper is organized as follows. Section 2 describes the baseline method and Section 3 illustrates the proposed extension in the masked CE loss. The experimental setup and results are presented in Section 4. We conclude the paper in Section 5.

## 2. THE BASELINE METHOD

In a lyrics-to-audio alignment system, lyrics are aligned with audio to produce time-synchronized lyrics. To do that, we first transform the musical audio into a spectrogram and convert the lyrics into a ground-truth phoneme sequence. Then, we build and train a model to predict phoneme probabilities for each frame based on the spectrogram. Finally, we align the model predictions with the ground-truth phoneme sequence to produce time-synchronized lyrics.

This section describes the baseline lyrics-to-audio alignment method, which is based on our previously proposed method in [25]. Readers are referred to [25] for further details.

### 2.1 Audio and Lyrics Pre-processing

For audio pre-processing, we first apply an advanced source separation method, HT Demucs [11], to extract the singing vocals. Then, we convert the extracted audio to a mono track and downsample it. We compute the log-scaled magnitude mel-spectrogram and normalize it to a range between 0 and 1. See Figure 3 for details.

For lyrics pre-processing, we convert the word sequence in the lyrics into a phoneme sequence. Word phonetization follows the conventions of the CMU Pronouncing Dictionary [1], which uses a set of 39 phonemes. In addition to these phoneme tokens, we add a token `0` to represent the blank symbol $\epsilon$ (staying in the same state) used in the CTC loss. We also add a token `40` for silence or space, resulting in 41 distinct tokens. Figure 4 shows an example of lyrics along with the corresponding phonemes and token representations.

[1] `http://www.speech.cs.cmu.edu/cgi-bin/cmudict`

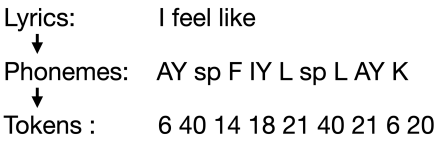



**Figure 4**: An example of converting lyrics to phonemes for lyrics pre-processing

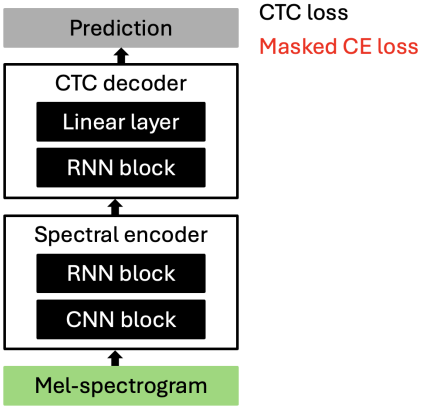



**Figure 5**: Model architecture. The baseline model is trained with the CTC loss. We propose and compare different masked CE losses to improve alignment performance.

### 2.2 Model Architecture

We adopt a simple Convolutional Recurrent Neural Network (CRNN) based on the models used in [24, 25]. The network architecture is shown in Figure 5: the mel-spectrogram is fed into a spectral encoder and a CTC decoder to produce frame-wise phoneme probabilities as model predictions.

The spectral encoder consists of a CNN block and an RNN block. The CNN block contains two 2D convolutional layers with 16 and 32 filters, respectively. Both layers use $3 \times 3$ kernels and `ReLU` activation. A sequence of operations are applied after each convolutional layer, including batch normalization, a pooling operation that halves the feature dimension, and a 25% dropout. The RNN block contains two bi-directional LSTM layers with 512 units in each layer.

In the CTC decoder, there are an RNN block (identical to the one used in the spectral encoder) and a linear layer with an output dimension of 41 (corresponding to the number of tokens). We apply a `softmax` function on the model output to obtain frame-wise phoneme probabilities.

This network is a simplified version of the model used in [25], where a spectral decoder was stacked after the CTC decoder to reconstruct the spectrogram for computing a reconstruction loss. In the paper, we discard the spectral decoder to focus on the masked CE loss.

### 2.3 The CTC Loss

We predict the frame-wise phoneme probabilities $\mathbf{D} = \mathbb{P}(\hat{\mathbf{y}}|\mathbf{X})$ from the input mel-spectrogram $\mathbf{X} \in \mathbb{R}^{N_F \times T}$, where $N_F$ is the number of frequency bins, $T$ is the number of time frames, and $\hat{\mathbf{y}} = \{\hat{y}_t\}, t \in [0, \cdots, T-1]$ represents the predicted phoneme sequence. In the phoneme probabilities $\mathbf{D} \in \mathbb{R}^{N_T \times T}$ (where $N_T$ is the number of tokens), its element $D_{i,t} = \mathbb{P}(\hat{y}_t = i|\mathbf{X})$ denotes the probability of predicting token $i$ at time $t$. Since the frame-wise annotations are unavailable, it is not possible to train the model by minimizing the loss between the predicted and ground-truth phonemes at each frame. Instead, we minimize the loss between the predicted and ground truth phoneme sequences by using Connectionist Temporal Classification (CTC) loss [22].

Set the phoneme sequence of the ground-truth lyrics $\mathbf{y} = \{y(m)\}$, $m \in [0, \ldots, M-1]$, where $M$ is the length of the ground-truth phoneme sequence, $y(m) \in \mathcal{A}$, and $\mathcal{A}$ denotes the set of all possible phonemes. The predicted phoneme sequence $\hat{\mathbf{y}}$ is an extended sequence of the ground-truth phoneme sequence $\mathbf{y}$. There is no time synchrony between these two phoneme sequences; but there is order synchrony between them (the ground-truth and predicted phoneme sequences happen in the same order). Usually, each phoneme in $\mathbf{y}$ spans one or several frames in $\hat{\mathbf{y}}$ with $M \leq T$. To maximize the probability of the predicted phoneme sequence $\hat{\mathbf{y}}$, we train the learnable model parameters $\Theta$ under the CTC constraints of $\hat{\mathbf{y}} \in \mathcal{A} \cap \{\epsilon\}$, and $\mathcal{B}(\hat{\mathbf{y}}) = \mathbf{y}$,

$$\mathbb{P}(\mathbf{y}|\mathbf{X};\Theta) = \sum_{\hat{\mathbf{y}},\mathcal{B}(\hat{\mathbf{y}})=\mathbf{y}} \prod_{t=0}^{T-1} \mathbb{P}(\hat{y}_t|\mathbf{X};\Theta), \tag{1}$$

where $\mathcal{B}$ is a mapping function that collapses repeated labels and removes blank symbols $\epsilon$. For example, $\mathcal{B}(\epsilon aaa\epsilon\epsilon bb\epsilon\epsilon) = ab$. The CTC loss is defined as the negative log-likelihood of the above probability:

$$\mathcal{L}_{\text{CTC}}(\Theta) = -\log \mathbb{P}(\mathbf{y}|\mathbf{X};\Theta). \tag{2}$$

### 2.4 Post-processing

Before forced alignment, we remove the silent segments at the beginning and end of each song to improve performance and robustness. First, we sum the magnitude mel-spectrogram along the frequency axis and normalize it to a maximum value of 1. Then, we apply a threshold of 0.05 to identify frames above the threshold as non-silent frames. We take the first non-silent frame as the start point and the last non-silent frame as the end point. We extend both points by one second (63 frames) to ensure that all singing parts are included. We only use the audio segment between the start and end points for alignment.

We align the model predictions with the ground-truth phoneme sequence using the CTC segmentation, a forced alignment method adapted for CTC-based model predictions [26]. The CTC segmentation is based on dynamic programming, in which the probability of staying in the same state is represented by the probability of the blank symbol $\epsilon$ as the way used in the CTC loss. We use the PyTorch implementation [2] of the CTC segmentation for forced alignment.

For phoneme-level alignment, we detect phoneme boundaries from the predicted phoneme sequence obtained by forced alignment. The segmented phonemes are merged into words (or lines) for word-level (or line-level) alignment. In this paper, we produce word-level alignment results for evaluation.

[2] `https://pytorch.org/audio/stable/tutorials/forced_alignment_tutorial.html`

| Word | Phonemes | Phoneme labels | Onset | Offset |
|---|---|---|---|---|
| I | AY | 6 | 0.087 seconds | 0.184 seconds |
| feel | F IY L | 14 18 21 | 0.281 seconds | 0.377 seconds |
| like | L AY K | 21 6 20 | 0.474 seconds | 0.571 seconds |

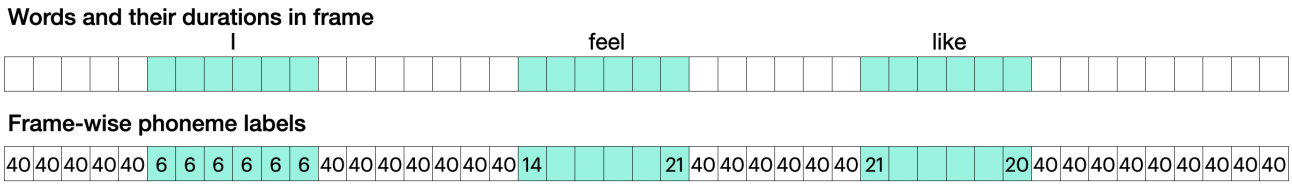



**Figure 6**: Word-level annotations and the derived phoneme-level labels at word onsets, word offsets, and silent frames.

## 3. EXPLORING MASKED CE LOSSES

Our work mainly focuses on using the masked CE loss along with the CTC loss to improve lyrics-to-audio alignment performance. Computing the masked CE loss includes two steps: (1) deriving frame-wise phoneme labels for key frames from word-level annotations and (2) constructing a mask on the key frames [25]. In this paper, the derived frame-wise phoneme labels in the first step remain the same as in our previous work [25]. Our extension lies in the second step: we propose four new masks and examine their impact on lyrics-to-audio alignment to identify effective masks for improving alignment performance.

### 3.1 The Derived Phoneme Labels for Key Frames

From weak annotations at the word level, we can derive frame-wise phoneme labels for specific frames, such as word onset and offset frames. For example, a musical clip with the lyrics "I feel like" would have word-level annotations that specify the onset and offset times of each word, as shown in Figure 6. From such annotations, we can derive phoneme labels for three types of frames:

1. For words containing two or more phonemes (e.g., "feel" and "like"), the first phoneme is assigned to the onset frame, and the last phoneme is assigned to the offset frame.

2. For words consisting of a single phoneme (e.g., "I"), all frames within the word duration are labeled with that phoneme.

3. For frames outside any word segments, a silence/space label (token `40`) is assigned.

### 3.2 Proposed Masks

These labeled frames (frames with derived labeled) play an important role in determining word onsets and offsets, which are essential for word-level alignment. To enforce the model prediction at these frames, we need to construct a mask to compute the masked CE loss on the masked frames.

In our previous work, we created a binary mask (mask_p) with ones at the labeled frames [25]. However, due to the introduction of the blank symbol $\epsilon$ in the CTC loss, non-onset frames can be represented by either phonemes or $\epsilon$, according to the CTC mapping function ($\mathcal{B}(\epsilon aaa\epsilon\epsilon bb\epsilon\epsilon) = ab$). In contrast, onset frames are always labeled by phonemes. To relax the constraint on non-onset frames, we set mask_p to zeros for $\epsilon$ (token 0) at these non-onset frames.

- mask_p: ones at all labeled frames, but zeros at non-onset frames for $\epsilon$ (token 0), as shown in Figure 7(b).

In this paper, we propose four new masks to explore the essential frames in the masks for performance improvements. However, we include $\epsilon$ for all masked frames in the new masks for two reasons: (1) the blank symbol $\epsilon$ is important in the CTC loss and post-processing; and (2) our previous results showed that the above mask (mask_p) does not help with word offset estimation. The proposed masks are listed as follows:

- mask_all: ones at all labeled frames, as shown in Figure 7(c). The masked CE loss is computed over all labeled frames to ensure that the model predicts the ground-truth phonemes at those frames.

- mask_onset: ones at word onset frames, as shown in Figure 7(d). The masked CE loss is computed over word onset frames to ensure that the model predicts the ground truth phonemes at onset frames.

- mask_onset_silence: ones at word onset frames and all silent frames, as shown in Figure 7(e). The masked CE loss is computed over word onset frames and all silent frames to improve silent frame detection for better word offset estimation.

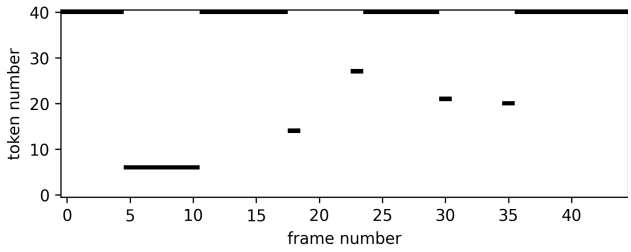
- mask_onset_slstart: ones at word onset frames and frames where silences begin, as shown in Figure 7(f). The masked CE loss is computed over word start frames and silence start frames, treating successive silent frames as a single segment.

### 3.3 The Masked CE loss

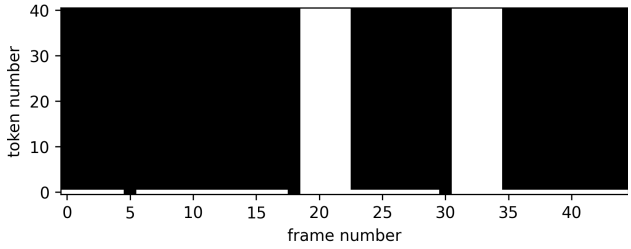
The masked CE loss is computed over the masked frames as follows:

$$\mathcal{L}_{\text{maskedCE}} = -\frac{1}{N_{\text{mask}}} \sum_{i} \sum_{t} B_{i,t} H(L_{i,t}, D_{i,t}), \qquad (3)$$

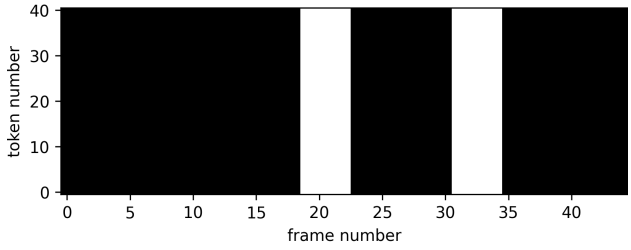
where $\mathbf{B} \in \mathbb{R}^{41 \times T}$ represents the mask and $\mathbf{L} \in \mathbb{R}^{41 \times T}$ represents the one-hot encoding of the derived phoneme



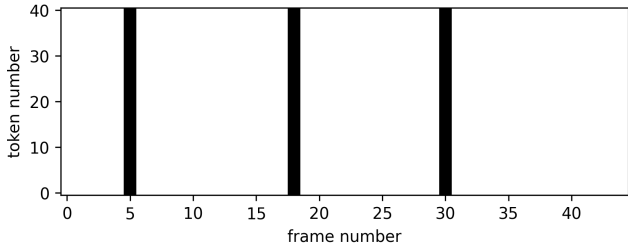

(a) The derived phoneme labels (one-hot encoding)



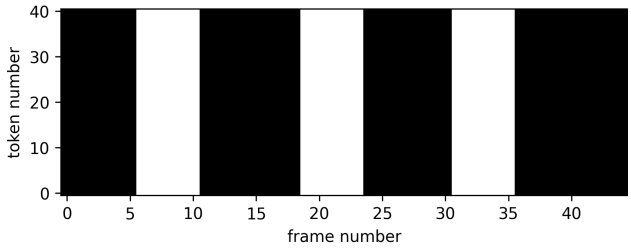

(b) Mask_p



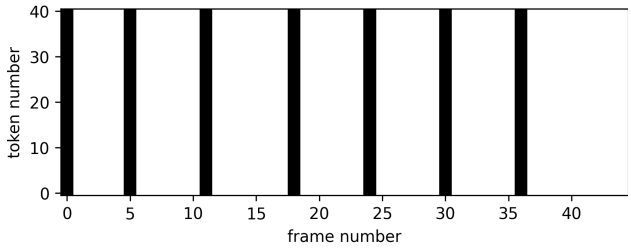

(c) Mask_all




(d) Mask_onset




(e) Mask_onset_silence




(f) Mask_onset_slstart

**Figure 7**: Derived frame-wise phoneme labels (one-hot encoding) and different masks, with black for ones and white for zeros. See Section 3.2 for mask details.

labels (see Figure 7 for examples). $\mathbf{D} \in \mathbb{R}^{41\times T}$ denotes the model predictions. $H(a,b) = a\log(b)$, and $N_{\text{mask}}$ is the total number of masked frames.

We can enforce the model predictions at the masked frames by applying the masked CE loss to assist training. In the experiments, the CRNN model is trained to minimize the following combined loss $\mathcal{L}$:

$$\mathcal{L} = \mathcal{L}_{\text{CTC}} + \lambda\mathcal{L}_{\text{maskedCE}}, \tag{4}$$

where $\lambda$ is the weight of the masked CE loss. We set $\lambda = 1$ for all the masks, as this value performed best in the preliminary experiment.

## 4. EXPERIMENTS AND RESULTS

### 4.1 Datasets

For training and validation, we use the DALI dataset [15], which contains over 5,000 songs along with annotations at four hierarchical levels: note, word, line, and paragraph. In our experiments, we use an English subset of the DALI dataset with 3,352 audio-available songs. This subset is randomly split into training and validation sets with an 80/20 ratio, resulting in 2,681 training songs and 671 validation songs.

To evaluate our method, we use the publicly available Jamendo dataset [23], which provides both line-level and word-level annotations. We present word-level results on 20 English songs in the Jamendo dataset and compare our results with other state-of-the-art (SOTA) results obtained on the same dataset.

During training and validation, the vocal audio is divided into 10-second segments with a 5-second hop size. For each segment, only the words that fall entirely within the segment are used. In contrast, during testing, the full vocal audio of each song is used without segmentation.

### 4.2 Training

We train the model for up to 20 epochs by using a batch size of 32 and the RMSprop optimizer with a learning rate of $10^{-4}$. To reduce variability caused by random initialization and dropout, each model is trained ten times with different random seeds. We report the results of all ten trained models, as well as their average.

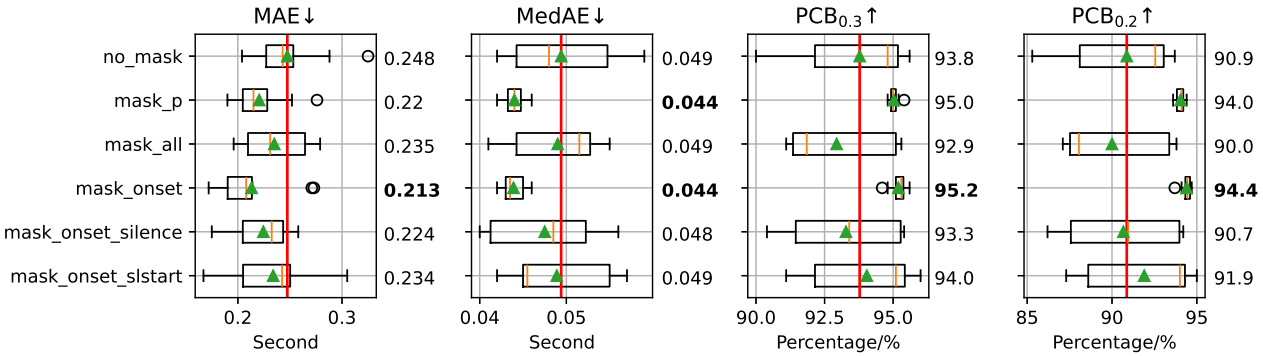
### 4.3 Evaluation Metrics

We conduct word-level evaluation by comparing the estimated word boundaries with the ground-truth word boundaries. The results are reported with different metrics, including: Mean Absolute Error (MAE) [27], Median Absolute Error (MedAE), and Percentage of Correct Boundaries (PCB) [3] with two tolerance windows [12].

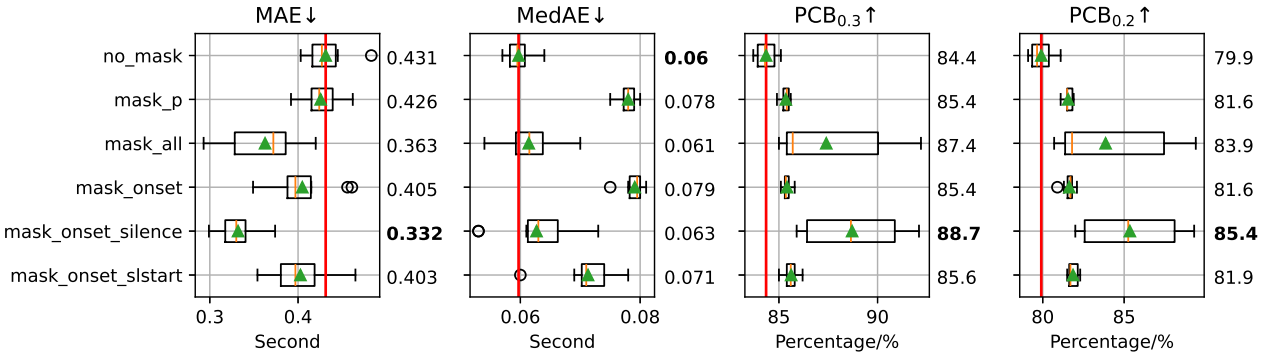
$$\text{MAE} = \frac{1}{W}\sum_{w=1}^{W}|t^w_{pred} - t^w_{ref}|, \tag{5}$$

$$\text{MedAE} = \underset{1\le w\le W}{\text{median}}(|t^w_{pred} - t^w_{ref}|), \tag{6}$$

[3] We use Percentage of Correct Boundaries (PCB) here in contrast to commonly-used Percentage of Correct Onsets (PCO) because we evaluate on two word boundaries (onsets and offsets).




(a) Word onset estimation results.




(b) Word offset estimation results.

**Figure 8**: Word-level lyrics-to-audio alignment results on the test Jamendo dataset. For each mask setting, the results of ten trained models are shown via a box-plot, with an orange line at the median and circles at fliers. The average value is indicated by the green triangle and also shown on the right. The baseline results (no_mask) are marked by red vertical lines for comparison. Bold numbers indicate the best results.

$$\text{PCB}_\tau = \frac{1}{W}\sum_{w=1}^{W} 1_{|t^w_{pred}-t^w_{ref}|<\tau} \times 100\%, \qquad (7)$$

where $W$ denotes the total number of words in a song, and $w$ denotes the word index. $t^w_{pred}$ and $t^w_{ref}$ represent the predicted and reference boundaries of the $w^{\text{th}}$ word, respectively. We compute PCB with two tolerance window sizes: $\text{PCB}_{0.3}$ and $\text{PCB}_{0.2}$, representing the percentage of correctly predicted word boundaries within 0.3 seconds and 0.2 seconds, respectively. We evaluate both word onsets and word offsets with results averaged over songs.

There is another commonly used evaluation metric: Percentage of Correct Segments (PCS), referring to the percentage of correct duration in the total duration [28]. This metric should consider both word onsets and offsets when determining the correct duration. However, in the practical, PCS is computed using only word onsets, with a word offset indicated by the onset of the next word [21, 18, 29]. Since this onset-only version of PCS does not help evaluate word offset estimation, we omit this metric in this paper.

### 4.4 Lyrics-to-Audio Alignment Results

Figure 8 illustrates the lyrics-to-audio alignment results of our proposed method, evaluated on word onsets and offsets, respectively. The results demonstrated that the masks had different effects on the estimation of word onsets and offsets. From Figure 8(a), we found that mask_onset performed best for word onset estimation with an MAE of 0.213 seconds. Mask_p followed closely behind with an MAE of 0.22 seconds.[4] By comparing the masks, we found that, for word onset estimation, masks on word onsets were more important than masks on other frames.

On the other hand, mask_onset_silence yields the best word offset estimation with an MAE of 0.332 seconds, as shown in Figure 8(b). Mask_all achieved the second best word offset estimation results with an MAE of 0.363 seconds. These results suggested that masks covering all silent frames were more effective for word offset estimation. By comparing word offset estimation results obtained by using mask_onset and mask_onset_slstart, we found that adding silence start frames in the mask (mask_onset_slstart) improved offset estimation results, while the MAE remained around 0.4 seconds. These suggested masks covering only silence start frames were insufficient to improve word offset estimation, whereas applying a strong constraint by masking all silent frames proved necessary.

Since we want to improve word offset estimation without compromising onset estimation performance, we combine the best onset results obtained by using mask_onset and the best offset results obtained by using mask_onset_silence as the results of the proposed method, as shown in Figure 2(b). The combined results are used for comparison in Table 1.

[4] For a fair comparison, we use the results obtained with the masked CE loss and without the reconstruction loss in [25] as the results of Mask_p.

| | Onset | | | | Offset | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Method | MAE ↓ | MedAE ↓ | $PCB_{0.3}$ ↑ | $PCB_{0.2}$ ↑ | MAE ↓ | MedAE ↓ | $PCB_{0.3}$ ↑ | $PCB_{0.2}$ ↑ | Training Dataset |
| GC [17] | 0.22 | 0.05 | 94% | | | | | | DALI |
| HBE [9] | 0.23 | | 94% | | | | | | DALI |
| KGLW [19] | 0.20 | | 94% | | | | | | DALI |
| HX-D [18] | 0.42 | **0.043** | 89% | 87% | | | | | DALI |
| HX-IH [18] | 0.16 | **0.043** | 93% | 91% | | | | | In-house 67k |
| DSE [23] | **0.15** | | 92% | | | | | | In-house 88k |
| Mask_p [25] | 0.220 | 0.044 | 95.0% | 94.0% | 0.426 | 0.078 | 85.4% | 81.6% | DALI |
| Baseline (no_mask) | 0.248 | 0.049 | 93.8% | 90.9% | 0.431 | **0.060** | 84.4% | 79.9% | DALI |
| Proposed | 0.213 | 0.044 | **95.2**% | **94.4**% | **0.332** | 0.063 | **88.7**% | **85.4**% | DALI |

**Table 1**: Comparison with SOTA methods. For the proposed method, the onset results are obtained by using mask_onset and the offset results are obtained by using mask_onset_silence. Bold numbers indicate the best results across all methods, while underlined numbers indicate the best results among methods trained using the DALI dataset.

### 4.5 A Comparison to SOTA Methods

Although our main goal is to improve word offset estimation in lyrics-to-audio alignment, we did not find any direct word offset estimation results from previous research. We present our word offset estimation results as a benchmark for future research and can only conduct a comparison on word onset estimation as follows.

Table 1 shows word onset estimation results of our proposed method (with mask_onset) and other SOTA lyrics-to-audio alignment methods. In comparison to other models trained on the DALI dataset, our model produced comparable results with an MAE of 0.213 seconds. While better MAEs were reported by models trained on large-scale in-house datasets: MAEs of 0.15 seconds and 0.16 seconds by DSE [23] and HX-IH [18], respectively. With respect to $PCB_{0.3}$, and $PCB_{0.2}$ metrics, our method outperformed all other methods, including those trained on in-house datasets.

## 5. CONCLUSIONS

In this paper, we extended our previous work of applying the masked CE loss to improve CTC-based lyrics-to-audio alignment. The extension aimed to enhance word offset estimation by exploring different masks in the masked CE loss. By comparing the results of using different masks, we identified the essential frames in the masks for improving word offset estimation and onset estimation: masks covering all silent frames were beneficial for word offset estimation; while masks covering word onset frames were essential for improving onset estimation. Our word onset estimation results were comparable to the SOTA results obtained by models trained on the same DALI dataset. Our word offset estimation results set a benchmark for future research since no existing results were found. In the future, we plan to explore phoneme-level lyrics-to-audio alignment and investigate the effectiveness of the masked CE loss on phoneme-level alignment.

### Acknowledgments

This work was supported in part by JST CREST Grant Number JPMJCR20D4 and JST CRONOS Grant Number JPMJCS25K1, Japan. We used ABCI 3.0 provided by AIST and AIST Solutions with support from "ABCI 3.0 Development Acceleration Use."

## 6. REFERENCES

[1] "MIREX2024: Lyrics-to-Audio Alignment," https://music-ir.org/mirex/wiki/2024:Lyrics-to-Audio_Alignment, accessed: 2025-12-10.

[2] H. Fujihara and M. Goto, "Lyrics-to-Audio Alignment and its Application," in *Multimodal Music Processing*, ser. Dagstuhl Follow-Ups. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2012, vol. 3, pp. 23–36. [Online]. Available: https://drops.dagstuhl.de/entities/document/10.4230/DFU.Vol3.11041.23

[3] J. Kato, T. Nakano, and M. Goto, "TextAlive: Integrated Design Environment for Kinetic Typography," in *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (ACM CHI 2015)*, 2015, pp. 3403–3412.

[4] A. M. Kruspe, "Bootstrapping a system for phoneme recognition and keyword spotting in unaccompanied singing," in *Proceedings of the 17th International Society for Music Information Retrieval Conference (ISMIR 2016)*, 2016, pp. 358–364.

[5] E. J. Humphrey, S. Reddy, P. Seetharaman, A. Kumar, R. M. Bittner, A. Demetriou, S. Gulati, A. Jansson, T. Jehan, B. Lehner, A. Krupse, and L. Yang, "An Introduction to Signal Processing for Singing-Voice Analysis: High Notes in the Effort to Automate the Understanding of Vocals in Music," *IEEE Signal Processing Magazine*, vol. 36, no. 1, pp. 82–94, 2019.

[6] B. Sharma, C. Gupta, H. Li, and Y. Wang, "Automatic Lyrics-to-audio Alignment on Polyphonic Music Using Singing-adapted Acoustic Models," in *Proceedings of the 2019 IEEE International Conference on Acoustics, Speech, and Signal Processing (IEEE ICASSP 2019)*, 2019, pp. 396–400.

[7] A. Vaglio, R. Hennequin, M. Moussallam, G. Richard, and F. d'Alché Buc, "Multilingual lyrics-to-audio alignment," in *Proceedings of the 21th International Society for Music Information Retrieval Conference (ISMIR 2020)*, 2020, pp. 512–519.

[8] E. Demirel, S. Ahlbäck, and S. Dixon, "Low Resource Audio-to-Lyrics Alignment from Polyphonic Music

Recordings," in *Proceedings of the 2021 IEEE International Conference on Acoustics, Speech, and Signal Processing (IEEE ICASSP 2021)*, 2021, pp. 586–590.

[9] J. Huang, E. Benetos, and S. Ewert, "Improving lyrics alignment through joint pitch detection," in *Proceedings of the 2022 IEEE International Conference on Acoustics, Speech, and Signal Processing (IEEE ICASSP 2022)*, 2022, pp. 451–455.

[10] J.-Y. Wang, C.-I. Leong, Y.-C. Lin, L. Su, and J.-S. R. Jang, "Adapting Pretrained Speech Model for Mandarin Lyrics Transcription and Alignment," in *Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU 2023)*, 2023.

[11] S. Rouard, F. Massa, and A. Défossez, "Hybrid Transformers for Music Source Separation," in *Proceedings of the 2023 IEEE International Conference on Acoustics, Speech, and Signal Processing (IEEE ICASSP 2023)*, 2023.

[12] M. Mauch, H. Fujihara, and M. Goto, "Integrating additional chord information into HMM-based lyrics-to-audio alignment," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 1, pp. 200–210, 2012.

[13] K. Schulze-Forster, C. S. J. Doire, G. Richard, and R. Badeau, "Phoneme Level Lyrics Alignment and Text-Informed Singing Voice Separation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 2382–2395, 2021.

[14] 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 (KDD 2020)*, 2020, pp. 1979–1989.

[15] G. Meseguer-Brocal, A. Cohen-Hadria, and G. Peeters, "DALI: a large Dataset of synchronized Audio, LyrIcs and notes, automatically created using teacher-student machine learning paradigm," in *Proceedings of the 19th International Society for Music Information Retrieval Conference (ISMIR 2018)*, 2018, pp. 431–437.

[16] ——, "Creating DALI, a large dataset of synchronized audio, lyrics, and notes," *Transactions of the International Society for Music Information Retrieval (TISMIR)*, vol. 3, no. 1, pp. 55–67, 2020.

[17] C. Gupta, E. Yılmaz, and H. Li, "Automatic Lyrics Alignment and Transcription in Polyphonic Music: Does Background Music Help?" in *Proceedings of the 2020 IEEE International Conference on Acoustics, Speech, and Signal Processing (IEEE ICASSP 2020)*, 2020, pp. 496–500.

[18] M. Kang, S. Park, and K. Choi, "HCLAS-X: Hierarchical and Cascaded Lyrics Alignment System Using Multimodal Cross-Correlation," arXiv 2307.04377, 2023. [Online]. Available: https://arxiv.org/abs/2307.04377

[19] T. Kick, F. Grötschla, L. A. Lanzendörfer, and R. Wattenhofer, "Contrastive Lyrics Alignment with a Timestamp-Informed Loss," in *Proceedings of the NeurIPS Workshop on AI-Driven Speech, Music, and Sound Generation*, 2024.

[20] C. Gupta, E. Yılmaz, and H. Li, "Acoustic modeling for automatic lyrics-to-audio alignment," in *Proceedings of the 20th Annual Conference of the International Speech Communication Association (Interspeech 2019)*, 2019, pp. 2040–2044.

[21] D. Stoller, S. Durand, and S. Ewert, "End-to-end lyrics alignment for polyphonic music using an audio-to-character recognition model," in *Proceedings of the 2019 IEEE International Conference on Acoustics, Speech, and Signal Processing (IEEE ICASSP 2019)*, 2019, pp. 181–185.

[22] A. Graves, S. Fernández, F. J. Gomez, and J. A. Schmidhuber, "Connectionist Temporal Classification: labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of the 23rd International Conference on Machine Learning (ICML 2006)*, 2006, pp. 369–376.

[23] S. Durand, D. Stoller, and S. Ewert, "Contrastive learning-based audio to lyrics alignment for multiple languages," in *Proceedings of the 2023 IEEE International Conference on Acoustics, Speech, and Signal Processing (IEEE ICASSP 2023)*, 2023.

[24] Y. Teytaut and A. Roebel, "Phoneme-to-Audio Alignment with Recurrent Neural Networks for Speaking and Singing Voice," in *Proceedings of the 22th Annual Conference of the International Speech Communication Association (Interspeech 2021)*, 2021, pp. 61–65.

[25] T. Cheng, T. Nakano, and M. Goto, "Improving lyrics-to-audio alignment using frame-wise phoneme labels with masked cross entropy loss," in *Proceedings of the 28th International Conference on Digital Audio Effects (DAFx25)*, 2025.

[26] L. Kürzinger, D. Winkelbauer, L. Li, T. Watzel, and G. Rigoll, "CTC-Segmentation of Large Corpora for German End-to-End Speech Recognition," in *Speech and Computer*, A. Karpov and R. Potapova, Eds., 2020, pp. 267–278.

[27] A. Mesaros and T. Virtanen, "Automatic alignment of music audio and lyrics," in *Proceedings of the 11th International Conference on Digital Audio Effects (DAFx08)*, Espoo, Finland, 2008.

[28] H. Fujihara, M. Goto, J. Ogata, and H. Okuno, "LyricSynchronizer: Automatic Synchronization System Between Musical Audio Signals and Lyrics," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 6, pp. 1252 – 1261, 2011.

[29] "MIREX2024: Lyrics-to-Audio Alignment Results," https://music-ir.org/mirex/wiki/2024:Lyrics-to-Audio_Alignment_Results, accessed: 2025-12-10.