# Singing MIDI Transcription with Music Language Models: Formulation and Comparison

Yu Sugimoto*, Jun-You Wang†, Li Su‡, and Eita Nakamura*§
* Kyushu University, Japan
† National Taiwan Normal University, Taiwan
‡ Academia Sinica, Taiwan
§ PTNA Institute of Music Research, Japan
E-mail: sugimoto.yu.681@s.kyushu-u.ac.jp, jywang@csie.ntnu.edu.tw,
lisu@iis.sinica.edu.tw, nakamura@inf.kyushu-u.ac.jp

*Abstract*—**This study investigates the use of music language models (LMs) in singing MIDI transcription, the task of estimating the pitch, onset time, and offset time of each note in the vocal part from a musical audio signal. While recent studies have investigated acoustic models that predict pitch frame by frame using deep neural networks (DNNs), transcription errors remain due to large pitch fluctuations and ambiguous note boundaries in singing. To address this issue, we formulate Markov- and DNN-based LMs that estimate pitch probabilities at the note level, and integrate them with a DNN-based acoustic model using two methods: generative modeling and the sequential transducer. Experimental results show that both integration methods significantly improve transcription accuracy over a baseline acoustic model. Moreover, different strengths and characteristics of the compared LMs and integration methods are discussed.**

## I. Introduction

Singing MIDI transcription, a task of automatic music transcription (AMT), is the problem of extracting the pitch, onset time, and offset time of each note of the singing part from a music audio signal. It is a foundational and challenging information processing technique for music analysis and retrieval [1]. An effective approach to this problem, as well as related tasks of singing F0 extraction [2] and singing score transcription [3], is the application of deep learning, which has progressively achieved increasing accuracies [4]–[7]. However, the current state-of-the-art methods [5]–[7] still suffer from estimation errors due to significant pitch variations, ambiguous note boundaries, and the presence of chorus parts in singing voices. Another challenge of the task is that a relatively small amount of training data can be used for research, which is different from the situation of analogous tasks such as piano transcription [8] and automatic speech recognition (ASR) [9].

A solution to reduce estimation errors is to use language models (LMs) that allow the calculation of the prior probability of output symbols, i.e. musical notes in music transcription and words in ASR. As they can incorporate prior knowledge or regularities in the output sequence, previous studies have reported successful results by integrating music LMs for music transcription tasks [10]–[12]. This study's purpose is to examine the potential of music LMs for singing MIDI transcription.

To integrate music LMs with deep neural network (DNN)-based transcription method, a new formulation is necessary for singing MIDI transcription because we require strict onset and offset times of pitches and rests unlike in the problem of ASR or music score transcription. As reviewed in Sec. II, there are several candidates in the types of LMs and in their integration methods, each exhibiting distinct advantages and limitations. As note-level music LMs, we formulate Markov model and recurrent neural network (RNN)-based model that can be efficiently trained using symbolic musical score data. We formulate integration methods based on two major approaches used for ASR, namely, generative modeling [13] and the sequential transducer [14], [15]. Based on these formulations, we construct several transcription algorithms and quantitatively compare their performances through evaluation experiments, drawing insights into the effectiveness of LM integration.

Our contributions are summarized as follows.

- General formulations for integrating music LM for monophonic MIDI transcription that can be applied to integrate a wide class of LMs and a wide class of DNN-based transcription methods.
- Empirical results providing insights for the potentials of the generative modeling and transducer approaches.

## II. Background: Language Model Integration for AMT and ASR

### A. Types of Music Language Models

Since music exhibits diverse forms and representations, various types of music LMs have been employed for AMT. Regarding the temporal units used for language modeling, frame-level units were adopted in [16], [17], tatum-level units in [11], [18], and note-level units in [3], [12]. While the use of frame- and tatum-level units enables the modeling of polyphonic music in a general framework, it poses challenges for capturing musically meaningful structures beyond repetitive pitch patterns [18]. Accordingly, note-level LMs are generally considered more effective for monophonic music transcription.

Another important aspect of music LMs is their architectural design. Markov models have been employed for monophonic modeling [3], [12], while deep generative models such as RNNs have been applied to polyphonic modeling [16], [17]. Given a sufficient amount of training data, DNN-based LMs can often achieve strong predictive performance.

Given this background, we investigate note-level LMs constructed using both Markov and DNN-based models. To the best of our knowledge, the integration of a note-level DNN-based LM for singing transcription or other AMT has not been explored previously.

### *B. Methods of Language Model Integration*

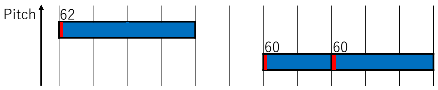
In AMT or AST, a frame-level recognition model (referred to as acoustic model) is typically constructed and integrated with a LM. Two major approaches to LM integration, originally developed for ASR, are generative modeling [13] and the sequential transducer [14]. In the former method, a probabilistic model is formulated to represent the generative process of acoustic features, and the LM is incorporated as a prior distribution over output symbols. During inference, output symbols are obtained by maximizing the posterior probability. In the latter method, both the output of the acoustic model and the predictions from the LM are fed into a DNN that estimates the final output symbols.

The generative modeling approach is most effective when used with a Markov LM, as the optimal sequence of output symbols can be efficiently computed using the Viterbi algorithm. For DNN-based LMs, approximate inference methods such as beam search are typically employed. A limitation of this approach is that it often requires a customized model formulation for each acoustic model, depending on its output type. In contrast, the transducer approach is applicable to a wider range of LMs and acoustic model architectures. However, a drawback of this method is that it does not permit exact inference over the entire output sequence; instead, approximate methods such as greedy or beam search must be used during inference. Since the optimal method for LM integration in AMT remains unclear, we investigate both of the two approaches in this study.

## III. Proposed Method

### *A. Problem Specification*

We aim to estimate the musical note sequence of a singing part from a musical audio signal. We first apply Demucs v4 [19] to extract the singing voice. Then, we obtain mel-spectrograms with a hop length of 10 ms from both the original signal and the separated singing signal, which are denoted as $[x_{tdf}]_{t=1,f=1}^{T,F}$, where $d$ indexes the channel (1 for the original mixture, 2 for the singing part), $T$ is the number of frames, and $F(=128)$ is the number of frequency bins. The input to the proposed system is the combination of the two mel-spectrograms, denoted as $\boldsymbol{X} := [\boldsymbol{x}_t]_{t=1}^{T} \in \mathbb{R}^{T\times 2\times F}$, where $\boldsymbol{x}_t := [x_{tdf}]_{d=1,f=1}^{2,F}$. The output note sequence is represented as $(t_l^{\mathrm{on}}, t_l^{\mathrm{off}}, p_l)_{l=1}^{L}$, where $t_l^{\mathrm{on}}$ and $t_l^{\mathrm{off}}$ are the onset and offset times, and $p_l \in \{0,\ldots,127\}$ is the pitch (MIDI note number) of the $l$-th note. $L$ denotes the number of notes.



| $t$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $\tilde{p}_t$ | 62 | 62 | 62 | 62 | 128 | 128 | 60 | 60 | 60 | 60 | 60 |
| $p'_t$ | 62 | 129 | 129 | 129 | 128 | 128 | 60 | 129 | 60 | 129 | 129 |
| $b_t$ | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| $c_t$ | 1 | 2 | 3 | 4 | 1 | 2 | 1 | 2 | 1 | 2 | 3 |
| $l(t)$ | 1 | 1 | 1 | 1 | 1 | 1 | 2 | 2 | 3 | 3 | 3 |

Fig. 1. Data Representation: frame index $t$, pitch including rest $\tilde{p}_t$, raw predicted label $p'_t$, onset label $b_t$, counter $c_t$, framewise note index $l(t)$.

### *B. Acoustic Model*

The acoustic model takes mel-spectrograms $\boldsymbol{X}$ as input and outputs the predicted probabilities $\boldsymbol{a}_t^{\mathrm{pitch}} = [a_{t\tilde{p}}^{\mathrm{pitch}}]_{\tilde{p}=0}^{128}$ for pitch and $\boldsymbol{a}_t^{\mathrm{onset}} = [a_{tb}^{\mathrm{onset}}]_{b=0}^{1}$ for onset label at each frame $t$:

$$a_{t\tilde{p}}^{\mathrm{pitch}} = P(\tilde{p}_t = \tilde{p}|\boldsymbol{X}), \quad a_{tb}^{\mathrm{onset}} = P(b_t = b|\boldsymbol{X}). \tag{1}$$

Here, the frame-wise pitch $\tilde{p}_t \in \{0,\ldots,128\}$ represents either a normal pitch $\tilde{p}_t < 128$ or a rest $\tilde{p}_t = 128$ (see Fig. 1). The onset label is defined as $b_t = 1$ if frame $t$ contains a note onset and $b_t = 0$ otherwise. The dimension of the final output $\boldsymbol{h}_t = \boldsymbol{a}_t^{\mathrm{pitch}} \oplus \boldsymbol{a}_t^{\mathrm{onset}}$ is $H = 129 + 2$. For the acoustic model, we use a convolutional RNN (CRNN) [20], which combines a convolutional neural network (CNN) with a bidirectional RNN containing LSTM units. The model is trained using the cross-entropy (CE) loss.

### *C. Music Language Model*

As for the LM, we employ an autoregressive generative model for the sequence of pitches $p_{1:L}$:

$$P(p_{1:L}) = \prod_{l=1}^{L} P_{\mathrm{lang}}(p_l|p_{1:(l-1)}). \tag{2}$$

Rests are not considered here because they are not relevant to the desired effect of the LM during transcription. The LM operates at the note level; at each step $l$, it takes $p_{l-1}$ as input and outputs the predicted probability $\boldsymbol{a}_l^{\mathrm{lang}} = [a_{lp}^{\mathrm{lang}}]_{p=0}^{127}$ for the next symbol $p_l$:

$$a_{lp}^{\mathrm{lang}} = P_{\mathrm{lang}}(p_l = p|p_{1:(l-1)}). \tag{3}$$

The dimension of the output $\boldsymbol{g}_t = \boldsymbol{a}_{l(t-1)+1}^{\mathrm{lang}}$ is 128, where $l(t)$ represents the index of the note sequence at time $t$.

To efficiently train the LM from musical score data in various pitch ranges and keys, we consider a model symmetric with respect to pitch transposition. First, the pitch $p_l$ is separated into a local tonic $s_l \in \{0,\ldots,11\}$ (corresponding to pitch class C in C major and A minor) and a relative pitch $p_l^{\natural} = p_l - s_l$. The relative pitch sequence $p_{1:L}^{\natural}$ is then represented as a sequence $q_{1:L}$ of extended pitch classes, which are invariant to octave transposition and retains a faithful representation of pitch intervals within 17 semitones downward and 18 semitones upward. The extended pitch class $q_l \in \{0,\ldots,35\}$ is defined as

$$q_1 = p_1^{\natural}\%12, \quad q_l = [(p_{l-1}^{\natural}\%12) + \mathrm{Clip}_{-17}^{18}(p_l^{\natural} - p_{l-1}^{\natural})]\%36,$$

where $\mathrm{Clip}_a^b(x)$ is a function that confines $x$ within the range $\{a, \ldots, b\}$ using a minimal number of octave shifts. The original pitch sequence $p_{1:L}$ can be recovered from $q_{1:L}$ using the following inverse transformation:

$$p_l = p_{l-1} - s_{l-1} + s_l + (q_l - q_{l-1}\%12 + 17)\%36 - 17. \quad (4)$$

We construct a generative model for the extended pitch classes $P_{\mathrm{lang}}^{\mathrm{epc}}(q_l = q|q_{1:(l-1)})$, and then the pitch-based LM symmetric with respect to pitch transposition can be obtained by

$$P_{\mathrm{lang}}(p_l|p_{1:(l-1)}, s_l) \propto \begin{cases} P_{\mathrm{uni}}^{\mathrm{pc}}(q_l), & l = 1; \\ P_{\mathrm{lang}}^{\mathrm{epc}}(q_l = q|q_{1:(l-1)}), & l \geq 2, \end{cases} \quad (5)$$

where $P_{\mathrm{uni}}^{\mathrm{pc}}(q_l)$ is the unigram probability of pitch classes. As concrete models, we construct two LMs: a first-order Markov model and a DNN-based model using a unidirectional LSTM network trained with the CE loss.

To apply the LM as a prior model of pitches in transcription, we need to estimate the tonic $s_l$ from the audio input. The method for this will be described in Sec. III-D.

### *D. Tonic Recognition*

For tonic recognition, we input 100 ms frame-level mel-spectrograms to a DNN that estimates the tonic $s_t \in \{0, \ldots, 11\}$ at each frame $t$. We train this DNN to output the predicted probability of tonic $s_t$, given by $\boldsymbol{a}_t^{\mathrm{tonic}} = [a_{ts}^{\mathrm{tonic}}]_{s=0}^{11}$:

$$a_{ts}^{\mathrm{tonic}} = P(s_t = s|\boldsymbol{X}). \quad (6)$$

The estimated tonic $\hat{s}_t$ is obtained by

$$\hat{s}_t = \underset{s}{\mathrm{argmax}}(a_{ts}^{\mathrm{tonic}}). \quad (7)$$

For tonic recognition, we use a CRNN with an architecture similar to that of the acoustic model (Sec. III-B). During transcription, the estimated tonic $\hat{s}_t$ is used to update the predicted probability $\boldsymbol{g}_t$ from the LM.

### *E. Integration Method 1: Generative Modeling*

In the generative modeling approach, we construct a probabilistic model for the note sequence $(t_l^{\mathrm{on}}, t_l^{\mathrm{off}}, p_l)_{l=1}^L$ and acoustic features $\boldsymbol{X} = [\boldsymbol{x}_t]_{t=1}^T$. To formulate the model, we convert the note sequence to its equivalent frame-level representation $(\tilde{p}_{1:T}, c_{1:T})$,where $\tilde{p}_t \in \{0, \ldots, 128\}$ denotes the (frame-level) pitch (128 denotes a rest) and $c_t \in \{1, \cdots, C\}$ denotes the counter of note duration (Fig. 1). $C$ is the maximum note duration. We assume the following factorization:

$$P(\tilde{p}_{1:T}, c_{1:T}, \boldsymbol{x}_{1:T}) = \prod_{t=1}^{T} P(\tilde{p}_t, c_t|\tilde{p}_{1:(t-1)}, c_{t-1}) P(\boldsymbol{x}_t|\tilde{p}_t, c_t). \quad (8)$$

This formulation resembles that of hidden semi-Markov model, where the first factor in the product represents the probability of hidden variables and the second the output probability.

The first factor in the right-hand side (RHS) of Eq. (8) can be transformed to the following form by introducing the onset variable $\tilde{b}_t \in \{0, 1\}$, which indicates the presence (1) or absence (0) of a note (here, including rest) onset as follows:

$$\begin{aligned} &P(\tilde{p}_t, c_t|\tilde{p}_{1:(t-1)}, c_{t-1}) \\ &\quad = \sum_{\tilde{b}_t \in \{0,1\}} P(\tilde{p}_t, c_t|\tilde{b}_t, \tilde{p}_{1:(t-1)}, c_{t-1}) P(\tilde{b}_t|\tilde{p}_{1:(t-1)}, c_{t-1}). \end{aligned}$$

The second factor in the RHS is related to the note duration probability, or equivalently, to the note exit probability $\pi_{\mathrm{exit}}(c)$ after a continuation of $c$ frames. Assuming that this factor is independent of the pitch, we have:

$$P(\tilde{b}_t|\tilde{p}_{1:(t-1)}, c_{t-1}) = \begin{cases} 1 - \pi_{\mathrm{exit}}(c_{t-1}), & \tilde{b}_t = 0; \\ \pi_{\mathrm{exit}}(c_{t-1}), & \tilde{b}_t = 1. \end{cases} \quad (9)$$

As a concrete form of the exit probability, we use an inverse gamma distribution

$$\pi_{\mathrm{exit}}(c) \propto \mathrm{IG}(c; \alpha, \beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} \frac{e^{-\beta/c}}{c^{\alpha+1}}, \quad (10)$$

which is empirically known to approximate the data distribution by fitting the shape and scale parameters, $\alpha$ and $\beta$.

If note continues ($\tilde{b}_t = 0$), then $c_t = c_{t-1}+1$ and $\tilde{p}_t = \tilde{p}_{t-1}$:

$$P(\tilde{p}_t, c_t|\tilde{b}_t = 0, \tilde{p}_{1:(t-1)}, c_{t-1}) = \delta_{\tilde{p}_t, \tilde{p}_{t-1}} \delta_{c_t, c_{t-1}+1}. \quad (11)$$

If note transits ($\tilde{b}_t = 1$), $c_t = 1$ and $\tilde{p}_t$ is generated according to the LM $\widetilde{P}_{\mathrm{lang}}(\tilde{p}_t|\tilde{p}_{1:(t-1)})$ extended to include rests:

$$P(\tilde{p}_t, c_t|\tilde{b}_t = 1, \tilde{p}_{1:(t-1)}, c_{t-1}) = \delta_{c_t,1} \widetilde{P}_{\mathrm{lang}}(\tilde{p}_t|\tilde{p}_{1:(t-1)}). \quad (12)$$

Here, the extended LM is constructed from the LM $P_{\mathrm{lang}}(p_l|p_{1:(l-1)})$ for normal pitches in Eq. (2) as

$$\begin{aligned} &\widetilde{P}_{\mathrm{lang}}(\tilde{p}_t|\tilde{p}_{1:(t-1)}) \\ &= \begin{cases} p_{\mathrm{rest}}^{(1)}, & \tilde{p}_t = 128,\ \tilde{p}_{t-1} \neq 128; \\ p_{\mathrm{rest}}^{(2)}, & \tilde{p}_t = 128,\ \tilde{p}_{t-1} = 128; \\ [1 - p_{\mathrm{rest}}^{(1)}] P_{\mathrm{lang}}(\tilde{p}_t|p_{1:l(t-1)}), & \tilde{p}_t \neq 128,\ \tilde{p}_{t-1} \neq 128; \\ [1 - p_{\mathrm{rest}}^{(2)}] P_{\mathrm{lang}}(\tilde{p}_t|p_{1:l(t-1)}), & \tilde{p}_t \neq 128,\ \tilde{p}_{t-1} = 128, \end{cases} \end{aligned} \quad (13)$$

where $p_{\mathrm{rest}}^{(1)}$ and $p_{\mathrm{rest}}^{(2)}$ represents the transition probability to a rest after a normal pitch and after a rest, respectively. With a Markov LM, this model for $P(\tilde{b}_t|\tilde{p}_{1:(t-1)}, c_{t-1})$ reduces to a semi-Markov model.

The output probability $P(\boldsymbol{x}_t|\tilde{p}_t, c_t)$ can be transformed to

$$\frac{P(c_t|\tilde{p}_t, \boldsymbol{x}_t) P(\tilde{p}_t|\boldsymbol{x}_t) P(\boldsymbol{x}_t)}{P(\tilde{p}_t, c_t)} \propto P(c_t|\tilde{p}_t, \boldsymbol{x}_t) P(\tilde{p}_t|\boldsymbol{x}_t), \quad (14)$$

where we have factored out $P(\boldsymbol{x}_t)$ irrelevant for our statistical inference problem and assumed that $P(\tilde{p}_t, c_t)$ is a uniform distribution. We can use the outputs of the acoustic model for the calculation of each factor in the RHS. $P(\tilde{p}_t = \tilde{p}|\boldsymbol{x}_t)$ can be substituted by the predicted probability of pitch $a_{t,\tilde{p}}^{\mathrm{pitch}}$. If $\tilde{p}_t = 128$ (rest), the acoustic feature $\boldsymbol{x}_t$ corresponds to silence and there is no onset for any $c_t$. Thus, we set

$$P(c_t|\tilde{p}_t = 128, \boldsymbol{x}_t) = a_{t,0}^{\mathrm{onset}}. \quad (15)$$

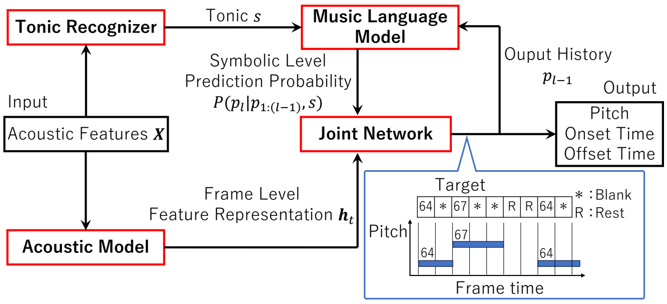


Fig. 2. Formulation of transducer-based method.

If $\tilde{p}_t$ is a normal pitch, there is an onset only when $c_t = 1$. Thus, we use the predicted probability for onset as follows:

$$P(c_t|\tilde{p}_t, \boldsymbol{x}_t) = \begin{cases} a^{\text{onset}}_{t,1}, & c_t = 1; \\ a^{\text{onset}}_{t,0}, & c_t \geq 2. \end{cases} \tag{16}$$

This completes the formulation of the generative modeling for $P(\tilde{p}_{1:T}, c_{1:T}, \boldsymbol{x}_{1:T})$.

The inference of the note sequence $(\hat{\tilde{p}}_{1:T}, \hat{c}_{1:T})$ given the inputs $\boldsymbol{x}_{1:T}$ can be expressed by

$$\begin{aligned} \hat{\tilde{p}}_{1:T}, \hat{c}_{1:T} &= \underset{p_{1:T}, c_{1:T}}{\operatorname{argmax}}\, P(\tilde{p}_{1:T}, c_{1:T}|\boldsymbol{x}_{1:T}) & (17) \\ &= \underset{p_{1:T}, c_{1:T}}{\operatorname{argmax}}\, P(\tilde{p}_{1:T}, c_{1:T}, \boldsymbol{x}_{1:T}). & (18) \end{aligned}$$

This can be computed using the Viterbi algorithm in the case of a Markov LM. In the case of a general LM, full optimization is computationally intractable, and beam search is applied.

Several refinements are introduced in the implementation. First, to control the relative influence of the LM, we introduce scaling factors $w_{\text{LM}}$ for the LM probability in Eq. (11) and $w_{\text{out}}$ for the pitch output probability in Eq. (14). Second, because the onset predicted probabilities by the acoustic model are typically low even at onsets, we introduce a factor $b_{\text{onset}}$ to amplify them.

### F. Integration Method 2: Transducer

A transducer [14], [15] is composed of an acoustic model, a LM, and a joint network. The joint network receives the acoustic model's output $\boldsymbol{h}_t$ and the LM's output $\boldsymbol{g}_t$ as input, and outputs the predicted probability $\boldsymbol{a}^{\text{joint}}_t = [a^{\text{joint}}_{tp'}]^{129}_{p'=0}$ for the set $\{0, \ldots, 129\}$ of pitches including the rest (128) and a blank symbol (129) at each time $t$:

$$a^{\text{joint}}_{tp'} = P(p'_t = p'|\boldsymbol{X}, p_{1:l(t-1)}). \tag{19}$$

For the training target, we use labels $p'_t \in \{0, \ldots, 129\}$ (called raw predicted labels), which include the blank to represent a continuation of note (Fig. 1). If a rest continues, we do not use the blank, to prevent the acoustic differences between notes and rests from negatively impacting training. The raw predicted sequence $p'_{1:T}$ can be interconverted with the frame-level pitch and onset labels $(\tilde{p}_{1:T}, b_{1:T})$ introduced in Sec. III-B.

During inference, the output sequence $\hat{p}'_t$ is estimated as

$$\hat{p}'_t = \underset{p'}{\operatorname{argmax}}\{a^{\text{joint}}_{tp'}\}. \tag{20}$$

The music LM receives the raw predicted label $p'_{t-1}$, which can include blanks and rests, as input and outputs the predicted probability $\boldsymbol{g}_t$ for each frame. If $p'_{t-1}$ is a blank or a rest, the predicted probability continues from the previous frame without an update. The state update of the music LM is represented by the following equation:

$$\boldsymbol{g}_t = \begin{cases} \boldsymbol{a}^{\text{lang}}_{l(t-1)+1}, & 0 \leq p'_t \leq 127; \\ \boldsymbol{g}_{t-1}, & \hat{p}'_t = 128 \text{ or } 129. \end{cases} \tag{21}$$

By utilizing blanks and rests in this manner, we determine the presence or absence of symbol-level transitions, thereby integrating frame-level and symbol-level information.

The CE loss function is adopted for training the joint network, which is constructed using a unidirectional LSTM network. Preliminary experiments revealed that training became difficult when using a 10 ms frame unit with this method. This is likely attributable to a significant increase in the proportion of blank and rest in the target labels. Consequently, the training and inference of the joint network are conducted at a 100 ms unit in our main experiment. Since a 100 ms unit for estimating onset and offset times is insufficient for transcription resolution, a conversion is performed from the 100 ms unit output to a 10 ms unit output.

## IV. Evaluation

### A. Experimental Setups

We constructed for the experiment a dataset (JBM) comprises 555 Japanese popular music songs, with audio and vocal tracks annotated in MIDI format. The reasons for using this data is its suitability for evaluation with realistic, commercial songs and the availability of large-scale music score data (used for training LMs) in the same musical genre. The dataset is randomly split into 331 songs for training, 112 for validation, and 112 for testing. For the training of the LMs, we employed our in-house data of the vocal-part musical scores of 5103 Japanese popular songs that have no overlap with the JBM data. To allow a reference to existing studies on singing MIDI transcription (e.g. [5]–[7]), we also used the MIR-ST500 dataset [21] for evaluation. Mel-spectrograms are obtained from acoustic signals resampled at 16 kHz using short-time Fourier transform and 128 mel-filter banks.

The acoustic model and tonic recognizer are constructed with identical CRNN architectures. The CNN portion of this CRNN follows the architecture proposed in [20]. For the RNN component, a 3-layer bidirectional LSTM is utilized, where the hidden layer dimension is set to 256. The LSTM LM is built using a 3-layer LSTM with a dimension of 256. The joint network is composed of a 4-layer LSTM with a dimension of 1024. During training, teacher forcing is applied with a probability of 80%. Since labels representing rest ($= 128$) and blank ($= 129$) appeared much more frequently than other labels in the target data, leading to data imbalance, the loss weights for those classes were set to 0.5. All these networks are trained using the Adam optimizer with a learning rate of $10^{-3}$. For the generative modeling, we used $\alpha = 3.683$,

$\beta = 0.798$, $p^{(1)}_{\text{rest}} = 0.001$, $p^{(2)}_{\text{rest}} = 0.001$, $w_{\text{LM}} = 1$, $w_{\text{out}} = 1$, and $b_{\text{onset}} = 5.88$, which were roughly optimized by several trials. $C$ was set to 300.

For evaluation, we use the COn, COnP, and COnPOff metrics [22], [23], which respectively measure the F1-score for (1) correct onset time, (2) correct onset time and pitch, and (3) correct onset time, pitch, and offset time. Following previous studies, we adopt the following criteria:

- **COn**: The absolute difference between the ground-truth and estimated onset times must be less than 50ms.
- **COnP**: The COn criterion must be satisfied, and the pitch must match the ground-truth.
- **COnPOff**: The COnP criterion must be satisfied, and the absolute difference between the ground-truth and estimated offset times must be less than $\max\{50 \text{ ms}, 0.2 \times$ (ground-truth note duration)$\}$.

### B. Experimental Results

To evaluate the performance of the two LMs, we calculated the CE using the JBM test set. The results were 2.858 (bits/symbol) for the Markov LM and 2.454 (bits/symbol) for the LSTM LM, clearly demonstrating the high predictive performance of the DNN-based LM.

Fig. 3 shows the average COnP F1-scores obtained by the generative modeling method. As the beam width increases, the F1-score rises monotonically and approaches the value obtained with Viterbi decoding. Under approximately optimal parameter settings ($w_{\text{LM}} = w_{\text{out}} = 1$), there is no notable difference between the two LMs. With an alternative setting ($w_{\text{LM}} = 0.5$, $w_{\text{out}} = 3$), the LSTM LM yielded slightly better performance. However, across all parameter configurations examined, the F1-score obtained by the LSTM LM with beam width 100 did not surpass that by the Markov LM using Viterbi decoding. This indicates that the Markov LM's advantage that full optimization of the output sequence can be achieved via Viterbi decoding outweighs the LSTM LM's advantage in providing more accurate pitch probability estimates.

Table I compares the performance of the generative modeling, transducer, baseline acoustic model (CRNN), a state-of-the-art singing MIDI transcription method, which was trained with a combination of the CTC and CE losses (denoted as "CTC&CE") [5][1]. For the generative modeling, the Markov LM and Viterbi decoding were used. Both the generative modeling and transducer significantly improved the COn and COnP F1-scores compared to the CRNN, clearly demonstrating the effectiveness of incorporating the LMs. The two transducers, which used different LMs, showed comparable performance, but both were outperformed by the generative modeling. The F1-scores by the generative modeling did not reach the CTC&CE method. In particular, the COnPOff F-scores for the proposed methods were substantially lower than that of the CTC&CE method, indicating further improvements in offset time estimation are needed. Similar tendencies were observed on the MIR-ST500 dataset.

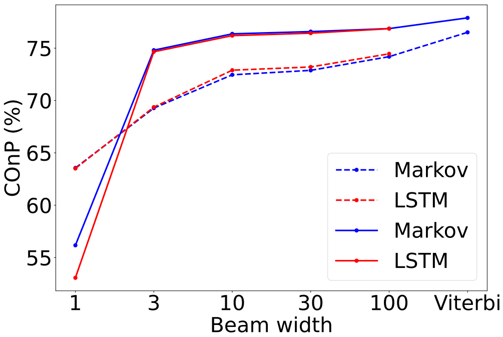


Fig. 3. COnP (%) for the generative modeling method with varying beam widths on the JBM data. The solid line shows the result for $w_{\text{LM}} = w_{\text{out}} = 1$ and the dotted line for $w_{\text{LM}} = 0.5$ and $w_{\text{out}} = 3$.

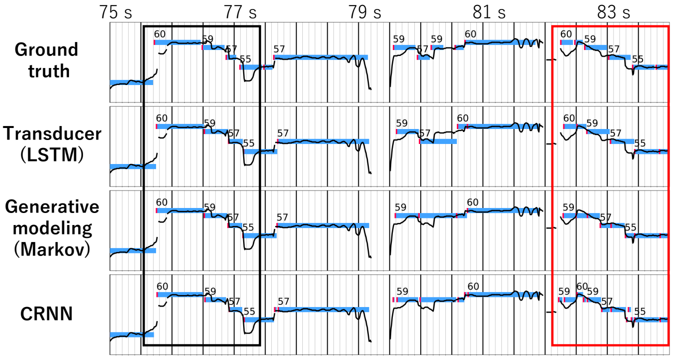


Fig. 4. Example transcription results. Note onsets are indicated by red lines and F0 contour is displayed in each panel.

Fig. 4 illustrates example transcription results[2]. In this example, due to fluctuations in the F0 contour, the CRNN produced numerous errors, including spurious very short notes. Many of these errors were corrected by the generative modeling; however, the first note in the right red box still exhibits a pitch error caused by a large F0 deviation. This error is resolved in the result by the transducer. Notably, this phrase consists of a repeated note pattern, as shown in the left black box, suggesting that the long-range memory of the transducer may have contributed to the correct transcription.

## V. Conclusions

We explored two approaches for integrating music LMs into singing MIDI transcription: generative modeling and the transducer. We examined the Markov model and the LSTM network as the LM. Our results demonstrated that all integration methods significantly improve transcription accuracy over the CRNN baseline. In particular, the generative modeling combined with Viterbi decoding achieved the best performance, benefiting from its ability to perform exact inference over the entire output sequence. With regard to LM comparison, the Markov and LSTM LMs showed comparable performance. While these results indicate that the classical hidden Markov model remains effective for the task, the transducer with the

[1] We retrained this model with the JBM dataset. The source code is available at: https://github.com/york135/CECTC_baseline_APSIPA25

[2] See also the demo page: https://ice.inf.kyushu-u.ac.jp/SMTwMLM/

TABLE I
COMPARATIVE EVALUATION RESULTS.

| Dataset | Method | COn (%) | | | COnP (%) | | | COnPOff (%) | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | Precision | Recall | F1-Score | Precision | Recall | F1-Score | Precision | Recall | F1-Score |
| JBM | CRNN | 75.18 | 87.14 | 80.37 | 68.39 | 79.16 | 73.06 | 50.93 | 58.78 | 54.36 |
| | Generative modeling (Markov) | 84.62 | 87.88 | 85.88 | 76.77 | 79.67 | 77.89 | 56.24 | 58.42 | 57.11 |
| | Transducer (LSTM) | 85.68 | 84.32 | 84.70 | 77.71 | 76.45 | 76.81 | 55.17 | 54.36 | 54.59 |
| | Transducer (Markov) | 84.34 | 86.00 | 84.87 | 76.33 | 77.79 | 76.80 | 54.57 | 55.60 | 54.91 |
| | CTC&CE loss [5] | 88.53 | 88.93 | 88.44 | 81.28 | 81.60 | 81.18 | 64.45 | 64.68 | 64.37 |
| MIR-ST500 | CRNN | 66.92 | 80.46 | 72.93 | 62.95 | 75.71 | 68.62 | 44.60 | 53.68 | 48.63 |
| | Generative modeling (Markov) | 76.26 | 80.33 | 78.10 | 71.86 | 75.75 | 73.62 | 49.05 | 51.67 | 50.23 |
| | Transducer (LSTM) | 76.56 | 80.13 | 78.19 | 71.82 | 75.21 | 73.36 | 48.46 | 50.64 | 49.46 |
| | Transducer (Markov) | 77.38 | 79.30 | 78.22 | 72.41 | 74.24 | 73.21 | 49.46 | 50.59 | 49.96 |
| | CTC&CE loss [5] | 80.28 | 79.24 | 79.66 | 75.12 | 74.23 | 74.58 | 58.10 | 57.49 | 57.72 |

LSTM LM performed better in cases where repeated note patterns were present. Although the proposed methods did not surpass the performance by the state-of-the-art methods [5], [6], the transducer has potential of further improvement by incorporating them as the acoustic model.

For future work, the transducer method can be expanded with the joint training of the acoustic model and LM. For the generative modeling, it is worth optimizing the balance between the acoustic model and LM and formulating a refined model for offset times. Since the repetitive patterns are frequent in musical rhythms, extending the proposed method for rhythm transcription is also worth investigating.

## ACKNOWLEDGMENT

This research was supported by JST FOREST JPMJPR226X and JSPS KAKENHI 25H01148 and 25H01169.

## REFERENCES

[1] M. Ryynanen and A. Klapuri, “Polyphonic music transcription using note event modeling,” in *IEEE WASPAA*, 2005, pp. 319–322.

[2] A. Jansson *et al.*, “Joint singing voice separation and f0 estimation with deep U-net architectures,” in *EUSIPCO*, 2019, pp. 325–329.

[3] T. Deng *et al.*, “End-to-end singing transcription based on CTC and HSMM decoding with a refined score representation,” *APSIPA TSIP*, vol. 13(5), no. e404, 2024.

[4] J.-Y. Hsu and L. Su, “VOCANO: A note transcription framework for singing voice in polyphonic music,” in *ISMIR*, 2021, pp. 293–300.

[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 TASLP*, vol. 31, pp. 383–396, 2022.

[6] J.-C. Wang *et al.*, “Mel-RoFormer for vocal separation and vocal melody transcription,” in *ISMIR*, 2024, pp. 454–461.

[7] L. Kim *et al.*, “Note-level singing melody transcription for time-aligned musical score generation,” *IEEE/ACM TASLP*, vol. 33, pp. 1088–1102, 2025.

[8] C. Hawthorne *et al.*, “Onsets and frames: Dual-objective piano transcription,” in *ISMIR*, 2018, pp. 50–57.

[9] R. Prabhavalkar *et al.*, “End-to-end speech recognition: A survey,” *IEEE/ACM TASLP*, vol. 32, pp. 325–351, 2023.

[10] A. Ycart and E. Benetos, “Polyphonic music sequence transduction with meter-constrained LSTM networks,” in *IEEE ICASSP*, 2018, pp. 386–390.

[11] R. Ishizuka *et al.*, “Tatum-level drum transcription based on a convolutional recurrent neural network with language model-based regularized training,” in *APSIPA ASC*, 2020, pp. 359–364.

[12] R. Nishikimi *et al.*, “Audio-to-score singing transcription based on a CRNN-HSMM hybrid model,” *APSIPA TSIP*, vol. 10, no. e7, 2021.

[13] G. Hinton *et al.*, “Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups,” *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[14] A. Graves, “Sequence transduction with recurrent neural networks,” in *preprint arXiv:1211.3711*, 2012.

[15] Y. He *et al.*, “Streaming end-to-end speech recognition for mobile devices,” in *IEEE ICASSP*, 2019, pp. 6381–6385.

[16] S. Sigtia *et al.*, “An end-to-end neural network for polyphonic piano music transcription,” *IEEE/ACM TASLP*, vol. 24, no. 5, pp. 927–939, 2016.

[17] A. Ycart *et al.*, “Blending acoustic and language model predictions for automatic music transcription,” in *ISMIR*, 2019, pp. 454–461.

[18] A. Ycart and E. Benetos, “A study on LSTM networks for polyphonic music sequence modelling,” in *ISMIR*, 2017, pp. 421–427.

[19] A. Défossez *et al.*, *Demucs(v4)*, https://github.com/facebookresearch/demucs [online], 2022.

[20] R. M. Bittner *et al.*, “Deep salience representations for F0 estimation in polyphonic music,” in *ISMIR*, 2017, pp. 63–70.

[21] J.-Y. Wang and J.-S. R. Jang, “On the preparation and validation of a large-scale dataset of singing transcription,” in *IEEE ICASSP*, 2021, pp. 276–280.

[22] C. Raffel *et al.*, “MIR_EVAL: A transparent implementation of common MIR metrics,” in *ISMIR*, 2014, pp. 367–372.

[23] E. Molina *et al.*, “Evaluation framework for automatic singing transcription,” in *ISMIR*, 2014, pp. 567–572.