JOINT ESTIMATION OF NOTE VALUES AND VOICES FOR AUDIO-TO-SCORE PIANO TRANSCRIPTION

Yuki Hiramatsu[1] **Eita Nakamura**[1,2] **Kazuyoshi Yoshii**[1,3]
[1]Graduate School of Informatics, Kyoto University, Japan
[2]The Hakubi Center for Advanced Research, Kyoto University, Japan
[3]PRESTO, Japan Science and Technology Agency (JST), Japan
{hiramatsu, enakamura, yoshii}@sap.ist.i.kyoto-u.ac.jp

ABSTRACT

This paper describes an essential improvement of a state-of-the-art automatic piano transcription (APT) system that can transcribe a human-readable symbolic musical score from a piano recording. Whereas estimation of the pitches and onset times of musical notes has been improved drastically thanks to the recent advances of deep learning, estimation of note values and voice labels, which is a crucial component of the APT system, still remains a challenging task. A previous study has revealed that (i) the pitches and onset times of notes are useful but the performed note durations are less informative for estimating the note values and that (ii) the note values and voices have mutual dependency. We thus propose a bidirectional long short-term memory network that jointly estimates note values and voice labels from note pitches and onset times estimated in advance. To improve the robustness against tempo errors, extra notes, and missing notes included in the input data, we investigate data augmentation. The experimental results show the efficacy of multi-task learning and data augmentation, and the proposed method achieved better accuracies than existing methods.

1. INTRODUCTION

The ultimate goal of automatic piano transcription (APT) is to convert a piano recording into a human-readable musical score that can be used for music analysis and performance [1]. This is a challenging task because of the polyphonic nature of piano music; musical notes form weakly-synchronous multiple streams called *voices* running in parallel. Much work on APT aims to estimate not a musical score but a *piano roll* from a music signal, *i.e.*, estimate the quantized pitches and non-quantized onset times of musical notes [2–7]. Although noticeable research progress has independently been made for multipitch detection [8–10] and rhythm transcription [11, 12], estimation of note values and voice labels, which is crucial for score typesetting with high readability [13], still remains a challenging task. Note that the note value represents the duration of a note on a symbolic musical score, and the voice label of a note specifies one of the voices of the upper or lower staff (right- or left-hand part) the note belongs to (Fig. 1).

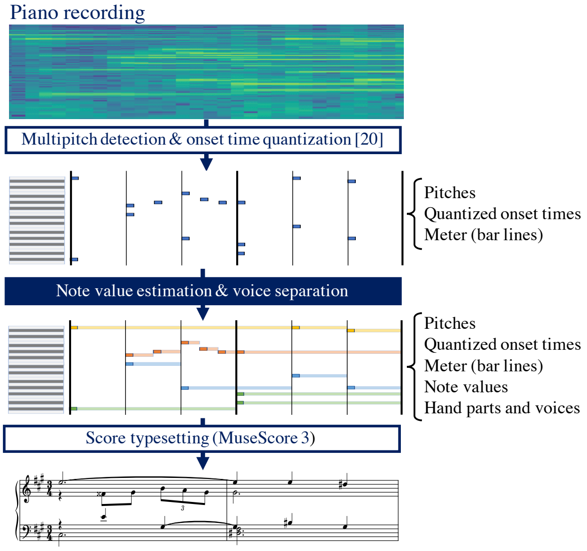


Figure 1. Overview of proposed method that jointly estimates note values and voice labels from transcribed pitches and onset times.

Several attempts have been made for estimating voice labels for piano scores having no voice labels [14–17]. Under an assumption that each voice has a strictly monophonic structure and note values are already transcribed with a certain degree of accuracy, voice labels can be estimated accurately [18, 19]. In practice, however, each voice has a homophonic structure consisting of concurrent notes (chords) and accurate estimation of note values is still an open problem. To deal with such a realistic situation, a state-of-the-art APT method uses a rule-based cost function with limited performance [20]. This calls for a principled statistical approach based on modern deep learning.

Note value estimation has relatively scarcely been studied [4, 21] and is still considered a challenging task [20]. Nakamura *et al.* [21] conducted a detailed statistical analysis and found that (i) the pitches and onset times of notes are useful but the performed note durations are less informative for estimating the note values and that (ii) the note

Attribution: Y. Hiramatsu, E. Nakamura, and K. Yoshii, “Joint Estimation of Note Values and Voices for Audio-to-Score Piano Transcription”, in *Proc. of the 22nd Int. Society for Music Information Retrieval Conf.,* Online, 2021.

values and voices have mutual dependency. The second conclusion derives from the fact that in a voice stream, the offset time of a note usually matches the onset time of the next note, or equivalently, short rests are rarely inserted between notes [4]. This indicates that joint estimation of note values and voice labels is an effective way of bringing improvements on both tasks.

In this paper, we propose a deep neural network (DNN) that estimates note values and voice labels jointly from a transcribed sequence of pitches and onset times for audio-to-score APT. Specifically, we train a bidirectional long short-term memory (BiLSTM) network in multi-task learning. We also investigate data representation, network architecture, post-processing, and data augmentation, which are considered to have an impact on the estimation performance. We report experimental evaluation conducted on datasets of classical and popular music, to investigate the efficacy of the joint estimation framework.

Our main contribution is to propose joint estimation of note values and voice labels based on deep learning. Combined with the state-of-the-art methods for multipitch detection and rhythm transcription used in the latest piano transcription system [20], we can achieve the state-of-the-art performance of audio-to-score APT. Another contribution is to propose new evaluation metrics for the polyphonic music transcription task, extending the one proposed in [22] to deal with voice labels. The example transcription results and the source code for the evaluation tool are available on the accompanying webpage [1] .

2. RELATED WORK

This section reviews methods for audio-to-score piano transcription, note value estimation, and voice separation.

2.1 Audio-to-Score Piano Transcription

Some piano transcription methods that can yield symbolic piano scores have been proposed, and the methods consisting of multi-stage processing [13, 20] achieved high accuracies. Cogliati *et al.* [13] proposed a transcription method that performs rhythm quantization and voice estimation for a piano performance MIDI file and generates a piano score. This method uses metrical, stream, and harmonic structures from the MIDI sequence estimated by a probabilistic model by Temperley [4]. Shibata *et al.* [20] proposed a state-of-the-art transcription method that can generate a piano score from audio signals with multi-stage processing. The method first estimates from a piano recording a performance MIDI sequence consisting of pitches, onset and offset times, and velocities using a convolutional neural network (CNN). The onset times are then quantized using a hidden Markov model (HMM). After note values and voice labels are separately estimated, piano scores are generated using MuseScore 3. In this study, we jointly estimate note values and voice labels, and aim to improve the accuracies that were lower than those of pitches and onset times in the method.

There are also end-to-end approaches that directly estimate musical scores from audio signals. Carvalho *et al.* [23] proposed a seq2seq model that estimates from an audio signal a piano score represented in the Lilypond music notation language. Román *et al.* [24] used a convolutional recurrent neural network (CRNN) that estimates a musical score represented in the **kern format. The network is trained with a connectionist temporal classification (CTC) loss function. These end-to-end methods have been tested only on very short or synthetic recordings, and there has been no account in the literature describing how well they perform in practice.

2.2 Note Value Estimation

Note value estimation is a difficult problem because note values do not always correspond to the performed durations [21]. Temperley [4] proposed a rhythm quantization method based on a probabilistic model. The method quantizes onset times by estimating beat positions. After voice labels are estimated, an offset time is set to the onset time of the next note in the same voice. One of the problems of the method is outputting no rests that are essential to make scores easy to read. Nakamura *et al.* [21] proposed a method based on Markov random fields. The method consists of a context model that represents a distribution of note values given pitches and onset times, and a performance model that generates actual performance durations from note values. It was shown that the performance model had a small impact on the estimation performance [21]. Therefore, we estimate note values only from pitches and onset times and do not use performed durations.

2.3 Voice Separation

Voice separation aims to divide musical notes into groups of notes representing musical streams. Karydis *et al.* [15] proposed a rule-based voice separation method for symbolic piano scores. The method is based on vertical integration, which integrates notes with the same onset time and the same duration, and horizontal integration, which integrates notes close in time and pitch. While this method can deal with homophonic voices, most other methods can only estimate monophonic voices. McLeod *et al.* [18] proposed a voice separation method for MIDI data using an HMM, and achieved high accuracy. Valk *et al.* [19] proposed a DNN-based voice separation method. The method uses a deep feedforward neural network that classifies each note represented by 33 handcrafted features into five classes. In piano transcription, these existing voice separation methods are not appropriate because voices often contain chords and durations are not estimated precisely. Explicit hand-part and voice labeling (rather than clustering) are also necessary for typesetting piano scores; for example, voice labels are used for determining the directions of note stems. Shibata *et al.* [20] proposed a cost-function-based voice separation method. Although this method is applicable to the situation of piano transcription, there is room for improvement in accuracy. We attempt to develop an improved DNN-based method.

[1] https://nvvest.github.io

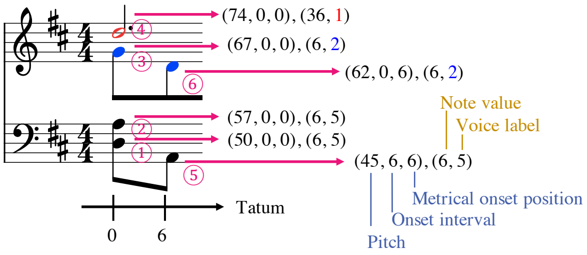


Figure 2. Data representation of the input and the output of the BiLSTM network.

3. PROPOSED METHOD

In this section, we propose BiLSTM networks that jointly estimate note values and voice labels, post-processing methods that correct the estimated note values, and data augmentation methods.

3.1 Problem Specification

Each note $\mathbf{z}_n = (p_n, o_n, d_n, v_n)$ in a musical score is represented by a pitch p_n, an onset time o_n, a note value d_n, and a voice label v_n. The pitch $p_n \in \{0, \dots, 127\}$ is represented by a MIDI note number (60 = C4). The onset time $o_n \geq 0$ and the note value $d_n \in \{0, \dots, 479\}$ are represented by integers (one measure is divided into 48 units); a zero note value is used for a grace note. Note that different meters have different tatum units: for example, a quarter note is represented as 12 in 4/4 time and 16 in 3/4 time. The maximum number of voices in each hand part is set to 4, following the convention for score notation used in score editing software such as MuseScore3 and Finale; labels $v_n = 1, 2, 3, 4$ are used for the right-hand part and $v_n = 5, 6, 7, 8$ for the left-hand part. We represent a piano score as a sequence of notes $\mathbf{Z} = \{\mathbf{z}_n\}_{n=1}^N$, where notes are arranged in the increasing order of onset times, and notes with the same onset time are ordered according to the pitches. Our goal is to estimate the note values and voice labels $\{(d_n, v_n)\}_{n=1}^N$ from a given set of pitches and onset times $\{(p_n, o_n)\}_{n=1}^N$.

3.2 BiLSTM Network

We propose a BiLSTM network that estimates note values and voice labels from pitches and onset times. We first represent the onset time o_n as the interval from the previous onset $i_n \in \{0, \dots, 767\}$ and the metrical position $b_n \in \{0, \dots, 47\}$ calculated as follows:

$$i_n = o_n - o_{n-1}, \quad b_n = o_n \bmod 48. \tag{1}$$

The input is then represented as $\mathbf{X} = \{(p_n, i_n, b_n)\}_{n=1}^N$ and the output is $\mathbf{Y} = \{(d_n, v_n)\}_{n=1}^N$ (Fig. 2).

The proposed network architecture is shown in Fig. 3(a). Each musical note of input $\mathbf{X}$ is represented as a $(128 \times 768 \times 48)$-dimensional one-hot vector. These one-hot vectors are first transformed to 25-dimensional feature vectors by a fully connected layer. The resulting vectors are then transformed to 50-dimensional vectors (latent representations) through a BiLSTM layer. Note value probabilities $\boldsymbol{\pi}_n(\mathbf{X}) = \{\pi_n(d; \mathbf{X})\}_{d=0}^{479}$ and voice label probabilities $\boldsymbol{\phi}_n(\mathbf{X}) = \{\phi_n(v; \mathbf{X})\}_{v=1}^{8}$ are separately calculated at each time step n after passing through fully connected layers and softmax layers, where $\pi_n(d; \mathbf{X})$ denotes the probability that the n-th note has duration d and $\phi_n(v; \mathbf{X})$ denotes the probability that the n-th note is in voice v.

We train the network by minimizing a cross-entropy loss function given by

$$\mathcal{L} = \mathcal{L}_{\mathrm{d}} + \mathcal{L}_{\mathrm{v}}, \tag{2}$$

where

$$\mathcal{L}_{\mathrm{d}} = -\sum_{n=1}^{N} \log \pi_n(d_n^*; \mathbf{X}), \tag{3}$$

$$\mathcal{L}_{\mathrm{v}} = -\sum_{n=1}^{N} \log \phi_n(v_n^*; \mathbf{X}), \tag{4}$$

where d_n^* and v_n^* are the correct note value and voice label, respectively. In the inference step, note values and voice labels are estimated from given pitches and onset times $\mathbf{X}$ as follows:

$$\hat{d}_n = \arg\max_{d} \pi_n(d; \mathbf{X}), \tag{5}$$

$$\hat{v}_n = \arg\max_{v} \phi_n(v; \mathbf{X}), \tag{6}$$

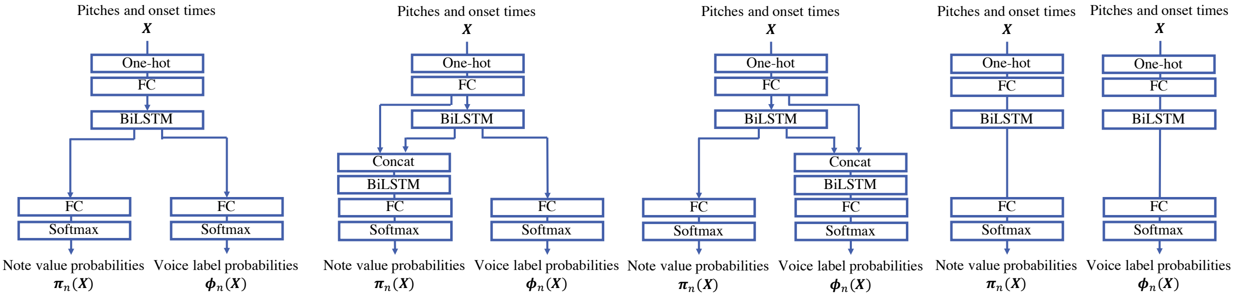
where $\hat{d}_n$ and $\hat{v}_n$ indicate the estimated note value and voice label, respectively.

3.3 Alternative Network Architectures

The network architecture in Fig. 3(a) is a simple joint network that equally treats the note value and voice label probabilities. We call this network SIM (simultaneous). We examine other network architectures shown in Fig. 3. As discussed in the Introduction, the voice structure has a strong impact on determining note values. To reflect this dependency structure, we propose the second network architecture (VLF; voice label first). In this network (Fig. 3(b)), voice labels are estimated first and note values are estimated with the latent representations used to estimate voice labels. For comparison, we also consider the third network architecture (NVF; note value first) that has a reverse structure (Fig. 3(c)). The networks SIM, VLF, and NVF are trained in a multi-task learning framework by minimizing the loss function $\mathcal{L}$ in Eq. (2). To confirm the efficacy of multi-task learning, we examine the fourth network architecture (IND; independent) that estimates note values and voice labels independently (Fig. 3(d)). IND consists of two BiLSTM networks and they are trained separately: by minimizing the loss functions $\mathcal{L}_{\mathrm{d}}$ and $\mathcal{L}_{\mathrm{v}}$, respectively. In all the network architectures, the first fully connected layer outputs 25-dimensional vectors, and each BiLSTM layer outputs a 50-dimensional hidden vector at each time step.

3.4 Post-Processing Methods

The note values and voice labels $\{(\hat{d}_n, \hat{v}_n)\}_{n=1}^N$ estimated by the network are sometimes inconsistent with the musical convention. As general rules, notes with the same onset time and the same voice should have the same note values. Also, the offset times of those notes should not be larger



(a) SIM (simultaneous) (b) VLF (voice label first) (c) NVF (note value first) (d) IND (independent)

Figure 3. Proposed BiLSTM networks for estimating note values and voice labels. The four architectures are explained in Section 3.3.

than the next onset time in that voice. For two notes n and m in the same voice, these constraints are represented as follows:

$$o_n = o_m \implies d_n = d_m, \tag{7}$$

$$o_n < o_m \implies d_n \leq o_m - o_n. \tag{8}$$

To impose the constraints, we consider three possible post-processing methods to adjust the estimated note values $\{\hat{d}_n\}_{n=1}^N$. Let $\{n_k\}_{k=1}^K$ index a set of notes with the same onset time in the same voice. In the first method (PP1), the note values $\{\hat{d}_{n_k}\}_{k=1}^K$ are all set to the interval to the next onset time as in [13]. Note that in this method note values are determined by the estimated voice labels and the note value probabilities are not used. In the second method (PP2), the note values are modified to their maximum value as follows:

$$\hat{d}'_{n_k} = \max_{l=1,\ldots,K} \hat{d}_{n_l}. \tag{9}$$

If the adjusted note values $\hat{d}'_{n_k}$ are longer than the interval to the next onset time, they are set to this interval. In the third method (PP3), we calculate the note value with the maximum probability from the candidate note values that satisfy the constraints as follows:

$$\hat{d}'_{n_k} = \mathop{\arg\max}_{d:\ d \leq d'} \prod_{l=1}^{K} \pi_{n_l}(d; \mathbf{X}), \tag{10}$$

where d' indicates the interval to the next onset time.

3.5 Data Augmentation

In the situation under consideration, the pitches and onset times used as the input $\mathbf{X}$ are estimated in advance by some pitch and rhythm transcription methods. As the result, the input contains tempo errors, extra notes, and missing notes. To make the networks robust to these errors, we can apply data augmentation methods that add tempo errors, extra notes, and missing notes to the original training data $\mathcal{D}$. Since rhythm transcription methods often produce half-tempo and double-tempo errors [20], we create a tempo-transformed dataset $\mathcal{D}_\mathrm{t}$ by halving or doubling the correct onset times and note values. Extra notes produced by multipitch detection methods often have a pitch shifted by an octave from a correct note. We thus create a dataset containing extra notes and missing notes $\mathcal{D}_\mathrm{em}$ by randomly deleting correct notes and adding notes whose pitches differ from correct pitches by one octave. In addition, to increase the amount of the training data, we implement another data augmentation method. Assuming that transposed piano scores are also musically valid, we train the network using data obtained by transposing the original data by an interval of δ semitones ($\delta = -12, -11, \ldots, 12$).

4. EVALUATION

We report experiments to evaluate the transcription accuracy of the proposed method.

4.1 Experimental Conditions

To evaluate the method in a practical condition, we incorporated it in an audio-to-score transcription system and generated transcriptions for test piano recordings. We first estimated pitches and quantized onset times by the state-of-the-art methods for multipitch detection and rhythm transcription used in the transcription system in [20]. For the results (called quantized MIDI data) we estimated note values and voice labels with the proposed method. We finally used public score editing software MuseScore 3 for score typesetting and generated transcriptions in the MusicXML format (Fig. 1). For comparison, we also generated transcriptions by existing methods [13, 20] using the same quantized MIDI data and with the same procedure for score typesetting. The CTD16 method [13] uses the Melisma Analyzer [4] for estimating note values and voice labels. The SNY21 method [20] is currently the best-performing system and uses a statistical model for note value estimation and a dynamic-programming method for voice separation.

As test data, we used 30 recordings of classical piano music in the MAPS-ENSTDkCl dataset [25] and 81 piano cover recordings of popular music used in [20]. The ground-truth musical scores for these recordings were prepared in the MusicXML format and used for assessing the generated transcriptions. We used the musical scores of 80 classical music pieces and 763 popular music pieces for training the BiLSTM networks; the same training data were used in [20]. We applied the data augmentation in Section 3.5 to these training samples.

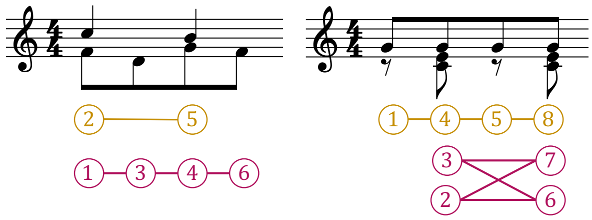


Figure 4. Voice structures represented by graphs.

4.2 Evaluation Metrics

We used the edit-distance-based error rates [22] for evaluating the quality of generated transcriptions. The error rates are the pitch error rate $\mathcal{E}_{\rm p}$, the missing note rate $\mathcal{E}_{\rm m}$, the extra note rate $\mathcal{E}_{\rm e}$, the onset-time error rate $\mathcal{E}_{\rm on}$, and the offset-time error rate $\mathcal{E}_{\rm off}$. They are calculated after aligning an estimated score with a correct score. In particular, we can use the offset-time error rate $\mathcal{E}_{\rm off}$ as a metric for evaluating the accuracy of estimated note values. Since these metrics do not evaluate the accuracy of estimated voice labels, we consider the voice error rate $\mathcal{E}_{\rm v}$ defined as the proportion of notes with incorrect voice labels. The overall error rate $\mathcal{E}_{\rm all}$ is defined as the mean of these six error rates.

The edit-distance-based metrics have a clear interpretation: they count how many notes or score elements should be edited to obtain the correct score. This is an advantage over other metrics for music transcription [26, 27]. We call the above defined metrics MUSTER (MUsic Score Transcription Error Rates) and the evaluation tool is made available online [2].

We also used an F-measure [17] for assessing the quality of estimated voice labels; this metric is conventionally used in studies on voice separation. The original metric [17] is formulated for monophonic voices and we here extend it for homophonic voices. A voice structure can be represented by a graph, where notes of consecutive chords in a voice are connected by an edge (Fig. 4). The graph can be represented by an adjacency matrix (a_{ij}), where $a_{ij} = 1$ when the i-th note is in a chord and the j-th note is in the next chord of the same voice, and otherwise $a_{ij} = 0$. We use the notation (a_{ij}) for a correct score and $(\hat{a}_{ij})$ for an estimated score. The precision $\mathcal{P}_{\rm v}$, the recall $\mathcal{R}_{\rm v}$, and the F-measure $\mathcal{F}_{\rm v}$ are defined as follows:

$$\mathcal{P}_{\rm v} = \frac{\sum_{i<j} a_{ij}\hat{a}_{ij}/\hat{w}_i}{\sum_{i<j} \hat{a}_{ij}/\hat{w}_i}, \quad \mathcal{R}_{\rm v} = \frac{\sum_{i<j} a_{ij}\hat{a}_{ij}/w_i}{\sum_{i<j} a_{ij}/w_i}, \tag{11}$$

$$\mathcal{F}_{\rm v} = \frac{2\mathcal{P}_{\rm v}\mathcal{R}_{\rm v}}{\mathcal{P}_{\rm v} + \mathcal{R}_{\rm v}}. \tag{12}$$

Here, $\sum_{i<j}$ signifies a summation over all notes i and all notes j that appear after i, and we have defined the weight for each note i as

$$w_i = \sum_{j>i} a_{ij}, \quad \hat{w}_i = \sum_{j>i} \hat{a}_{ij} \tag{13}$$

in order to normalize the contribution of each chord no matter how many notes it contains.

Method	$\mathcal{E}_{\rm off}$	$\mathcal{E}_{\rm v}$	$\mathcal{P}_{\rm v}$	$\mathcal{R}_{\rm v}$	$\mathcal{F}_{\rm v}$
SIM+DA	33.3	**39.1**	63.9	**64.9**	64.0
VLF+DA	**32.2**	**39.0**	**65.2**	**65.7**	**65.1**
NVF+DA	**32.9**	40.7	63.1	62.6	62.5
IND+DA	**32.9**	40.5	64.1	63.8	63.6
VLF	**33.1**	**39.1**	**64.3**	64.3	64.0

Table 1. Error rates (%) and accuracies (%) of estimated note values and voice labels for the MAPS dataset. DA indicates that each network is trained with augmented data.

Method	$\mathcal{E}_{\rm off}$	$\mathcal{E}_{\rm v}$	$\mathcal{P}_{\rm v}$	$\mathcal{R}_{\rm v}$	$\mathcal{F}_{\rm v}$
SIM+DA	**17.9**	12.2	**87.1**	**87.3**	**87.1**
VLF+DA	**17.2**	**11.4**	**87.7**	**87.7**	**87.6**
NVF+DA	**18.1**	12.4	**87.4**	**87.2**	**87.2**
IND+DA	18.7	12.5	**86.8**	86.4	86.5
VLF	**17.5**	**11.4**	**87.5**	**87.8**	**87.6**

Table 2. Error rates (%) and accuracies (%) of estimated note values and voice labels for the J-pop dataset.

Method	MAPS	J-pop
VLF+DA	32.2	17.2
VLF+DA+PP1	**28.0**	**15.3**
VLF+DA+PP2	31.4	16.3
VLF+DA+PP3	32.2	16.8

Table 3. Error rates $\mathcal{E}_{\rm off}$ (%) of estimated note values with different post-processing methods.

4.3 Experimental Results

We first compare the four network architectures (SIM, VLF, NVF, and IND) with or without the application of data augmentation (DA). The evaluation results are listed in Tables 1 and 2 for the MAPS dataset and the J-pop dataset, respectively. Among the four architectures trained with data augmentation, VLF achieved the best accuracy in both note values and voice labels. The higher accuracy of VLF compared to NVF indicates that it is better to estimate voice labels first. A comparison between VLF and IND confirms the efficacy of multi-task learning. By comparing the results for VLF with and without data augmentation, we found a positive effect of data augmentation. Similar results were obtained for the other network architectures.

We next compare the three post-processing methods (Table 3). The first method (PP1) achieved the lowest error rates. The second one (PP2) slightly reduced the offset error rates for both datasets. Before and after the third method (PP3), the error rates were almost the same. In the first post-processing method, note values are calculated from estimated voice labels and note value probabilities estimated by the network are not used. Importantly, this does not mean that note value estimation was useless in the present method: estimating note values by the network was effective for improving the voice estimation through the multi-task learning, which in turn led to more accurate note value estimations.

The first method also has a limitation that it cannot estimate rests. Rests are used to express articulations and to make scores easier to read. An example of the tran-

[2] https://amtevaluation.github.io/

Method	Test	$\mathcal{E}_\mathrm{p}$	$\mathcal{E}_\mathrm{m}$	$\mathcal{E}_\mathrm{e}$	$\mathcal{E}_\mathrm{on}$	$\mathcal{E}_\mathrm{off}$	$\mathcal{E}_\mathrm{v}$	$\mathcal{E}_\mathrm{all}$	$\mathcal{P}_\mathrm{v}$	$\mathcal{R}_\mathrm{v}$	$\mathcal{F}_\mathrm{v}$
Proposed (VLF+DA+PP1)	MAPS	**0.67**	**8.11**	**6.23**	**11.6**	**28.0**	**39.1**	**15.6**	**65.2**	**65.7**	**65.1**
SNY21 [20]	MAPS	**0.67**	**8.11**	**6.23**	**11.5**	**28.3**	44.6	16.6	62.4	59.4	60.6
CTD16 [13]	MAPS	**0.88**	13.5	**6.33**	16.8	44.0	74.3	26.0	56.0	42.5	47.9
Proposed (VLF+DA+PP1)	J-pop	**0.61**	**4.03**	**7.29**	**2.67**	**15.3**	**11.4**	**6.89**	**87.6**	**87.7**	**87.6**
SNY21 [20]	J-pop	**0.61**	**4.03**	**7.29**	**2.69**	20.9	18.0	8.92	78.6	77.0	77.7
CTD16 [13]	J-pop	**0.82**	12.8	**7.21**	8.48	55.7	65.8	25.1	51.3	38.8	44.0

Table 4. Error rates (%) and accuracies (%) of transcription. The CTD16 method could output results for 27 (72) pieces in the MAPS (J-pop) dataset; the metrics are calculated from these pieces.

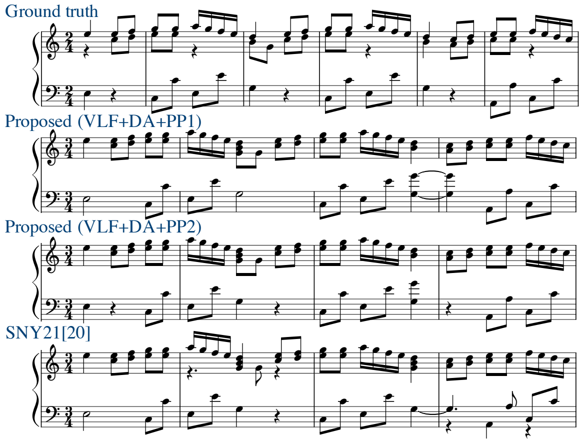


Figure 5. Example transcription results. The proposed method improved the accuracy of note values.

scription results is shown in Fig. 5, where the proposed method with the second post-processing method correctly estimated rests. In the future, it is important to estimate rests in order to improve the average accuracy of note values.

We finally compare the proposed method with existing transcription methods [13, 20]. The full set of MUSTER metrics and the voice F measure for the MAPS and J-pop datasets are listed in Table 4. The present method achieved the best accuracy for both datasets. The transcription accuracies for the MAPS dataset were lower than those for the J-pop dataset because the former has more complicated voice structures and there were a small number of classical music pieces in the training data.

To compare the performance of the voice estimation by our method with a recent method focusing on voice separation, we also evaluated the HMM-based voice separation method (MS16) [18]. Since this method requires as input pitches, onset times, and offset times, we used the note values estimated by the network IND. The F-measures $\mathcal{F}_\mathrm{v}$ of the voice separation results by MS16 were 55.7% and 66.5% for the MAPS dataset and the J-pop dataset, respectively. It is confirmed that the proposed method significantly outperformed the MS16 method.

An example of the transcription results is shown in Fig. 6, for the proposed method and the SNY21 method. The proposed method estimated voice labels close to the ground truth, and made the piano score easier to read than the one estimated by the SNY21 method. Other examples are shown on the supplemental web page [3] .

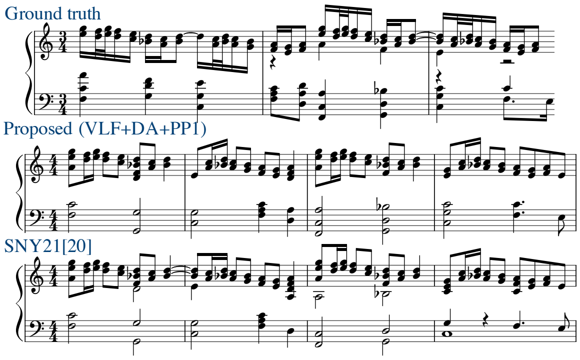


Figure 6. Example transcription results. The proposed method improved the accuracy of voice labels and improved the readability.

5. CONCLUSION

This paper presented a neural method that jointly estimates note values and voice labels from transcribed pitches and onset times. Since note values and voices are interrelated, we constructed a BiLSTM network in a multi-task learning framework. We demonstrated through experiments that the proposed method achieved the state-of-the-art performance of audio-to-score APT when combined with the latest methods for multipitch detection and onset time quantization.

The error rates of note values and voice labels are still high compared to the other metrics. In future work, we plan to further investigate the data representation and network architecture to increase the consistency between estimated note values and voice labels. To correctly estimate rests and improve the readability of transcribed scores, we will develop a more sophisticated post-processing method and study the effective use of performed durations.

Although we focused on the estimation of note values and voices in this study, we found that the result is affected by errors made by the onset time quantization method. It is thus important to develop a method that integrates onset time quantization. As the fully end-to-end approaches still have difficulties in practical applications [23, 24], it is also considered effective to unify the multiple stages in Fig. 1 one step after another.

6. ACKNOWLEDGEMENT

This work is supported in part by JST PRESTO No. JPMJPR20CB, JSPS KAKENHI Nos. 19H04137, 19K20340, 20K21813, and 21K12187, and ISHIZUE 2021 of Kyoto University Research Development Program.

[3] https://nvvest.github.io

7. REFERENCES

[1] E. Benetos, S. Dixon, Z. Duan, and S. Ewert, “Automatic music transcription: An overview,” *IEEE Signal Processing Magazine*, vol. 36, no. 1, pp. 20–30, 2018.

[2] P. Desain and H. Honing, “The quantization of musical time: A connectionist approach,” *Computer Music Journal*, vol. 13, no. 3, pp. 56–66, 1989.

[3] C. Raphael, “A hybrid graphical model for rhythmic parsing,” *Artificial Intelligence*, vol. 137, pp. 217–238, 2002.

[4] D. Temperley, “A unified probabilistic model for polyphonic music analysis,” *Journal of New Music Research*, vol. 38, no. 1, pp. 3–18, 2009.

[5] E. Vincent, N. Bertin, and R. Badeau, “Adaptive harmonic spectral decomposition for multiple pitch estimation,” *IEEE TASLP*, vol. 18, no. 3, pp. 528–537, 2010.

[6] E. Benetos and T. Weyde, “An efficient temporally-constrained probabilistic model for multiple-instrument music transcription,” in *ISMIR*, 2015, pp. 701–707.

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

[8] C. Hawthorne, E. Elsen, J. Song, A. Roberts, I. Simon, C. Raffel, J. Engel, S. Oore, and D. Eck, “Onsets and frames: Dual-objective piano transcription,” in *ISMIR*, 2018, pp. 50–57.

[9] Y.-T. Wu, B. Chen, and L. Su, “Polyphonic music transcription with semantic segmentation,” in *ICASSP*, 2019, pp. 166–170.

[10] Q. Kong, B. Li, J. Chen, and Y. Wang, “GiantMIDI-Piano: A large-scale MIDI dataset for classical piano music,” *arXiv preprint arXiv:2010.07061*, 2020.

[11] E. Nakamura, K. Yoshii, and S. Sagayama, “Rhythm transcription of polyphonic piano music based on merged-output HMM for multiple voices,” *IEEE/ACM TASLP*, vol. 25, no. 4, pp. 794–806, 2017.

[12] E. Nakamura and K. Yoshii, “Music transcription based on Bayesian piece-specific score models capturing repetitions,” *arXiv preprint arXiv:1908.06969*, 2019.

[13] A. Cogliati, D. Temperley, and Z. Duan, “Transcribing human piano performances into music notation.” in *ISMIR*, 2016, pp. 758–764.

[14] J. Kilian and H. H. Hoos, “Voice separation-A local optimization approach,” in *ISMIR*, 2002, pp. 39–46.

[15] I. Karydis, A. Nanopoulos, A. Papadopoulos, E. Cambouropoulos, and Y. Manolopoulos, “Horizontal and vertical integration/segregation in auditory streaming: A voice separation algorithm for symbolic musical data,” in *Proc. of Sound and Music Computing Conference*, 2007, pp. 299–306.

[16] E. Cambouropoulos, “Voice and stream: Perceptual and computational modeling of voice separation,” *Music Perception*, vol. 26, no. 1, pp. 75–94, 2008.

[17] B. Duane and B. Pardo, “Streaming from MIDI using constraint satisfaction optimization and sequence alignment.” in *Proc. of International Computer Music Conference*, 2009.

[18] A. McLeod and M. Steedman, “HMM-based voice separation of MIDI performance,” *Journal of New Music Research*, vol. 45, no. 1, pp. 17–26, 2016.

[19] R. de Valk and T. Weyde, “Deep neural networks with voice entry estimation heuristics for voice separation in symbolic music representations,” in *ISMIR*, 2018.

[20] K. Shibata, E. Nakamura, and K. Yoshii, “Non-local musical statistics as guides for audio-to-score piano transcription,” *Information Sciences*, vol. 566, pp. 262–280, 2021.

[21] E. Nakamura, K. Yoshii, and S. Dixon, “Note value recognition for piano transcription using Markov random fields,” *IEEE/ACM TASLP*, vol. 25, no. 9, pp. 1846–1858, 2017.

[22] E. Nakamura, E. Benetos, K. Yoshii, and S. Dixon, “Towards complete polyphonic music transcription: Integrating multi-pitch detection and rhythm quantization,” in *ICASSP*, 2018, pp. 101–105.

[23] R. G. C. Carvalho and P. Smaragdis, “Towards end-to-end polyphonic music transcription: Transforming music audio directly to a score,” in *Proc. of IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 2017, pp. 151–155.

[24] M. A. Román, A. Pertusa, and J. Calvo-Zaragoza, “A holistic approach to polyphonic music transcription with neural networks,” in *ISMIR*, 2019, pp. 731–737.

[25] V. Emiya, R. Badeau, and B. David, “Multipitch estimation of piano sounds using a new probabilistic spectral smoothness principle,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 6, pp. 1643–1654, 2009.

[26] A. Cogliati and Z. Duan, “A metric for music notation transcription accuracy.” in *ISMIR*, 2017, pp. 407–413.

[27] A. McLeod and M. Steedman, “Evaluating automatic polyphonic music transcription.” in *ISMIR*, 2018, pp. 42–49.