PHASE-AWARE JOINT BEAT AND DOWNBEAT ESTIMATION BASED ON PERIODICITY OF METRICAL STRUCTURE

Takehisa Oyama[1] **Ryoto Ishizuka**[1] **Kazuyoshi Yoshii**[1,2]
[1]Graduate School of Informatics, Kyoto University , Japan
[2]PRESTO, Japan Science and Technology Agency (JST), Japan
{ooyama, ishizuka, yoshii}@sap.ist.i.kyoto-u.ac.jp

ABSTRACT

This paper describes a phase-aware joint beat and downbeat estimation method mainly intended for popular music with a periodic metrical structure and steady tempo. The conventional approach to beat estimation is to train a deep neural network (DNN) that estimates the *beat presence probability* at each frame. This approach, however, relies heavily on a periodicity-aware post-processing step that detects beat times from the noisy probability sequence. To mitigate this problem, we have designed a DNN that estimates the *beat phase* at each frame whose period is equal to the beat interval. The estimation losses computed at all frames not limited to a fewer number of beat frames can thus be effectively used for backpropagation-based supervised training, whereas a DNN has conventionally been trained such that it constantly outputs zero at all non-beat frames. The same applies to downbeat estimation. We also modify the post-processing method for the estimated phase sequence. For joint beat and downbeat detection, we investigate multi-task learning architectures that output beat and downbeat phases in this order, in reverse order, and in parallel. The experimental results demonstrate the importance of phase modeling for stable beat and downbeat estimation.

1. INTRODUCTION

Rhythm analysis of music signals such as beat, downbeat, and tempo estimation often constitutes the crucial front end of automatic music transcription [1, 2] and music structure analysis [3]. The typical approach to beat estimation consists of (1) computing an onset strength signal (OSS) from a music signal and (2) detecting regularly-spaced beat times from the OSS with autocorrelation analysis or comb filtering [4–6]. The step (1) has recently been implemented with a deep neural network (DNN) that outputs the probability of the presence of a beat at each frame [7–11]. In particular, convolutional neural networks (CNNs) attained the noticeable improvement of beat estimation [7–9]. Since the same applies to downbeat estimation, we often focus on only beat estimation in the remainder of the paper.

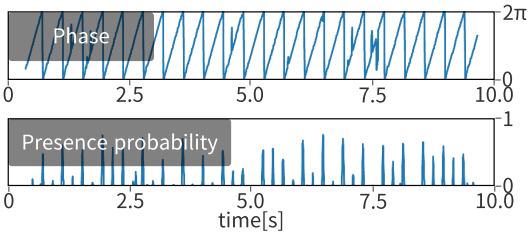

Figure 1. For beat estimation, the proposed DNN aims to estimate a sawtooth-shaped beat phase sequence, whereas a conventional DNN aims to estimates an impulsive beat presence probability sequence.

The major problem of the typical approach is that the periodic nature of beat times is not considered explicitly in the step (1). The performance of DNN-based beat estimation thus heavily depends on the periodicity-aware post-processing step (2) that detects beat times from the noisy probability sequence. This calls for the improved accuracy of the raw output of a DNN used in the step (1).

To solve this problem, in this paper we propose a new approach to beat estimation that aims to estimate not the *beat presence probability* but the *beat phase* at each frame (Fig. 1). Note that a sequence of beat phases is represented as a semi-continuous sawtooth wave whose period corresponds to beat intervals, whereas a sequence of beat presence probabilities as an impulse train that takes one at only beat frames and zero at the other frames. The key advantage of the phase-based representation is that all frames not limited to beat frames give meaningful information about the periodic beat structure. Since a DNN is usually trained with backpropagation such that the sum of frame-level estimation losses is minimized, the phase-based representation would be a more suitable target for periodicity-aware supervised training.

We also propose a post-processing method that detects beat times from the noisy phase sequence (Fig. 2). In recent beat estimation [7–10], a dynamic Bayesian network (DBN) based on the bar-pointer model [12], which is approximately implemented as a hidden Markov model (HMM) [13], is used for picking beat times from a number of peaks included in a beat probability sequence. We modify the observation model of the DBN to deal with a beat phase sequence. The global tempo can be estimated by identifying the most dominant frequency component from the Fourier transform of the noisy sinusoidal wave converted from the estimated phase sequence.

Since beat and downbeat times form the hierarchical metrical structure of music in a mutually dependent manner, we investigate three multi-task learning architectures for joint beat and downbeat estimation. More specifically, one can (1) predict beat phases from an audio spectrogram and then predict downbeat phases from the spectrogram and the estimated beat phases, (2) predict the downbeat and beat phases in this order (the reverse order of (1)), and (3) predict both the beat and downbeat phases in parallel. Several examples estimated by the proposed method are available at https://phase2bdbt.github.io/.

2. RELATED WORK

Classical beat estimation methods focus on the periodicity of a music signal with signal processing techniques [5, 6]. In recent years, DNNs have actively been used for directly estimating the probability of the presence of a beat at each frame, given the spectrogram or acoustic features of a music signal. The first attempt for DNN-based beat estimation used a long short-term memory (LSTM) network that estimates a sequence of beat presence probabilities from mel spectrograms with different window lengths [14]. More recently, the temporal convolutional network (TCN) [15] was shown to improve the performance with shorter training time [9]. It consists of dilated convolution layers like WaveNet [16] originally proposed for audio synthesis such that the receptive field of a deeper layer becomes wider exponentially to capture the long-term dependency of time-series data. For better estimation, the TCN used in [9] has a non-causal architecture, *i.e.*, both the past and future input data are used for making a prediction at the current frame, whereas the original TCN has a causal architecture.

For tempo estimation, one can estimate the tempo from a kind of onset strength signal (OSS) extracted from a music signal and detect the most dominant period corresponding to the tempo from the OSS with autocorrelation analysis, comb filtering, or the discrete Fourier transform (DFT) [17–19]. In the same way as beat estimation, the tempo has recently been estimated directly from a music signal with a DNN [20, 21], where the tempo estimation is interpreted as a classification problem. Schreiber *et al.* [20] attempted to estimate local tempos as well as the global tempo. Foroughmand *et al.* [21] proposed a new representation of the DNN input called harmonic constant-Q modulation (HCQM) that represents the harmonic series considering tempo frequencies.

Several multi-task methods have been proposed for joint estimation of mutually-dependent multiple kinds of metrical elements. In the earliest years, Goto [22] proposed a method based on signal processing techniques and expert knowledge of metrical structure for real-time joint beat and downbeat estimation. In recent years, the LSTM was used for joint beat and downbeat estimation [10] and the TCN was used for joint beat and tempo estimation [8], which was extended for joint beat, downbeat, and tempo estimation [7], resulting in the state-of-the-art performances. In [7], a single TCN was shared over three tasks and was trained with a data augmentation technique.

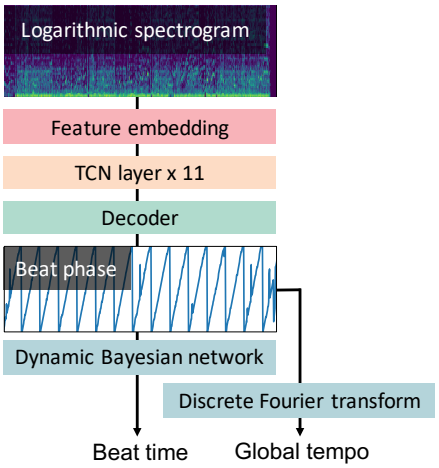

Figure 2. The proposed phase-aware beat estimation followed by tempo estimation.

3. PROPOSED METHOD

This section describes the proposed phase-aware method for rhythm analysis. Our goal is to estimate beat and downbeat times and the global tempo from the log-magnitude spectrogram $\mathbf{X} \in \mathbb{R}^{F \times T}$ of a music signal, where F is the number of frequency bins and T is the number of frames. For beat estimation, we perform DNN-based phase classification (Section 3.1) followed by DBN-based peak picking (Section 3.2) and tempo estimation (Section 3.3). We then describe three possible architectures of multi-task learning for joint beat and downbeat estimation (Section 3.4).

3.1 DNN-Based Beat/Downbeat Phase Classification

We tackle beat phase estimation in terms of a DNN-based classification problem. In our preliminary investigation on the Beatles dataset [23], we found that when a DNN is used for phase regression, it often fails to decrease the estimation loss without careful pretraining. The beat phase is reset to zero at a beat frame and linearly increases to 2π until the next beat frame, *i.e.*, the phase sequence forms a sawtooth wave (Fig. 1). The phase resolution is set to $2\pi/K$, *i.e.*, the phase is quantized into K classes.

Let $\mathbf{Z}^b \triangleq \{\mathbf{z}_t^b\}_{t=1}^T$ be a sequence of beat phases, where $\mathbf{z}_t^b \in \{0,1\}^K$ is a K-dimensional one-hot vector at frame t whose k-th element z_{tk}^b takes one when the beat phase z_t^b satisfies $\frac{2\pi(k-1)}{K} \leq z_t^b < \frac{2\pi k}{K}$. Let $\mathbf{Z}^d \triangleq \{\mathbf{z}_t^d\}_{t=1}^T$ be a sequence of downbeat phases defined in the same way. Hereafter, * is denoted as ^{b} or ^{d}. In practice, we use a *blurry* version of $\mathbf{Z}^*$ as target data for training a DNN-based classifier. More specifically, when $z_{tk}^* = 1$, we assume that $z_{t,k\pm1}^* = 0.75$, $z_{t,k\pm2}^* = 0.50$, and $z_{t,k\pm3}^* = 0.25$.

Let $\boldsymbol{\psi}^* \triangleq \{\boldsymbol{\psi}_t^*\}_{t=1}^T$ be a sequence of class probability vectors estimated by the DNN, where $\boldsymbol{\psi}_t^* \in [0,1]^K$ is a K-dimensional normalized vector of frame t. The DNN is trained in a supervised manner such that it maximizes the posterior probability of $\mathbf{Z}^*$ given by

$$\mathcal{J}_{\text{phase}}^* = \frac{1}{T}\sum_{t=1}^{T}\sum_{k=1}^{K} z_{tk}^* \log \psi_{tk}^*. \tag{1}$$

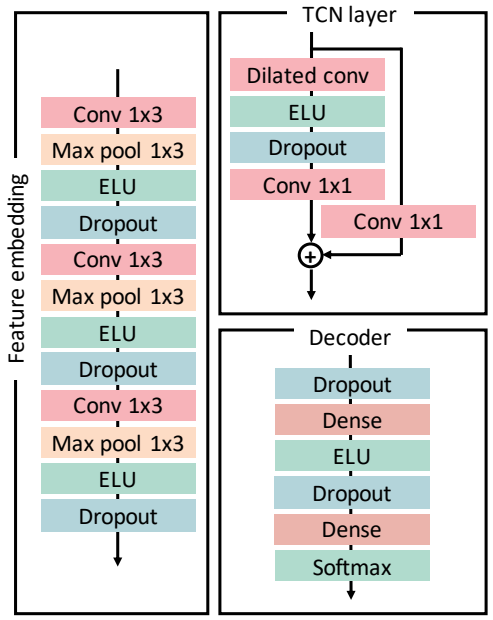

Figure 3. Network architectures.

The overall structure of the proposed method is shown in Fig. 2 and the detailed architecture of the DNN is shown in Fig. 3. The DNN used for phase classification in this study is basically the same as one used in the latest study [7] except that skip connections in the TCN used for tempo estimation are removed. It takes as input the log-magnitude spectrogram of a music signal on a logarithmic frequency axis, which is fed to the feature extraction layer referred to as *Feature embedding* in Fig. 2. The extracted feature vectors with 20 channels at each frame are fed to a stack of eleven TCN layers, which is referred to as *TCN layer* × *11*, followed by *Decoder* that outputs a sequence of beat or downbeat phases. For the K-class outputs, we slightly modify the components of *Decoder* as in Fig. 3. One can refer in particular to [9] for detailed descriptions of the other DNN components.

3.2 DBN-Based Beat/Downbeat Detection

We modify the existing dynamic Bayesian network (DBN) used in [13] for detecting beat and downbeat times from a noisy sequence of the estimated beat and downbeat phases. The main modification lies in the change of the observed variable of the DBN from the presence probabilities to the phases.

3.2.1 State Space

For each frame t, we represent the tempo S_t^v as the number of frames per beat, $\mathrm{S}_t^v \in \{s^v_{\min}, s^v_{\min}+1, s^v_{\min}+2, \ldots, s^v_{\max}\}$ ($\mathrm{S}_t^v \in \mathbb{Z}$) where $s^v_{\min}$ and $s^v_{\max}$ are calculated as follows:

$$s^v_{\min} = \left\lceil \frac{60 \times \mathrm{fps}}{\mathrm{BPM}_{\max}} \right\rceil, \; s^v_{\max} = \left\lceil \frac{60 \times \mathrm{fps}}{\mathrm{BPM}_{\min}} \right\rceil, \qquad (2)$$

where $\mathrm{BPM}_{\min}$ ($\mathrm{BPM}_{\max}$) indicates the minimum (maximum) BPM, fps indicates the number of frames per second, and $\lceil x \rfloor$ denotes the closest integer to x. Let BPB be the number of beats per measure, and $N_t = \mathrm{BPB} \times S_t^v$ is the number of frames per measure. Let $\mathrm{S}_t^p \in \{1, 2, \ldots, N_t\}$ be the position in a measure at frame t, and $\mathbf{S} \triangleq \mathbf{S}_{1:T} = (\mathbf{S}^p_{1:T}, \mathbf{S}^v_{1:T})$ be a sequence of hidden states. The hidden states at frame t is shown in Fig 4

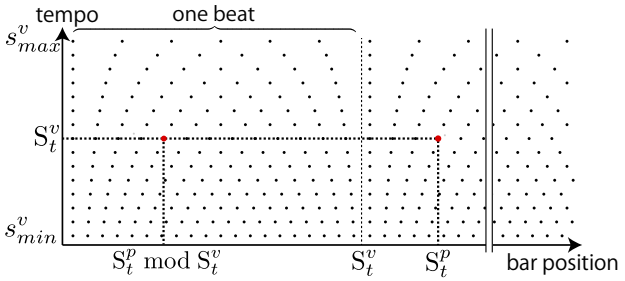

Figure 4. Two-dimensional representation of hidden state space at frame t used in DBN.

3.2.2 State Transition Model

We use the same transition model as [13], where the transition probabilities are computed as follows:

$$\begin{aligned} p(\mathbf{S}_t|\mathbf{S}_{t-1}) &= p(\mathrm{S}^p_t, \mathrm{S}^v_t|\mathrm{S}^p_{t-1}, \mathrm{S}^v_{t-1}) \\ &= p(\mathrm{S}^p_t|\mathrm{S}^p_{t-1}, \mathrm{S}^v_{t-1}) p(\mathrm{S}^v_t|\mathrm{S}^p_t, \mathrm{S}^v_{t-1}). \end{aligned} \qquad (3)$$

The first term of (3) represents a bar transition model, which is defined as

$$\begin{aligned} &p(\mathrm{S}^p_t|\mathrm{S}^p_{t-1}, \mathrm{S}^v_{t-1}) \\ &\quad = \begin{cases} 1 & (\mathrm{S}^p_t - 1 \equiv \mathrm{S}^p_{t-1} \bmod N_{t-1}); \\ 0 & (\text{otherwise}). \end{cases} \end{aligned} \qquad (4)$$

The second term of (3) represents a tempo transition model and the tempo is only allowed to change at beat times. $p(\mathrm{S}^v_t|\mathrm{S}^p_t, \mathrm{S}^v_{t-1})$ is defined as follows: if $\mathrm{S}^p_t \in \mathcal{B}$,

$$p(\mathrm{S}^v_t|\mathrm{S}^p_t, \mathrm{S}^v_{t-1}) = \exp\left(-\lambda \times \left|\frac{\mathrm{S}^v_t}{\mathrm{S}^v_{t-1}} - 1\right|\right), \qquad (5)$$

otherwise

$$p(\mathrm{S}^v_t|\mathrm{S}^p_t, \mathrm{S}^v_{t-1}) = \begin{cases} 1 & (\mathrm{S}^v_t = \mathrm{S}^v_{t-1}); \\ 0 & (\text{otherwise}), \end{cases} \qquad (6)$$

where $\mathcal{B}$ is the set of positions that corresponds to beats and $\lambda \in \mathbb{Z}_{\geq 0}$ is the parameter to determine the steepness of the above distribution.

3.2.3 Observation Model

We formulate an observation model that stochastically generates an acoustic feature sequence $\mathbf{X}$ from a latent state sequence $\mathbf{S}$. We use the output of the DNN as the probability distribution of a certain phase sequence $\mathbf{Z}^*$ given the acoustic features $\mathbf{X}$ as follows:

$$p(\mathbf{Z}^*|\mathbf{X}) = \prod_{t=1}^{T} \prod_{k=1}^{K} (\psi^*_{tk})^{z^*_{tk}}. \qquad (7)$$

We represent $\mathbf{Z}^b$ and $\mathbf{Z}^d$ together as $\mathbf{Z} = (\mathbf{Z}^b, \mathbf{Z}^d)$. To compute $p(\mathbf{X}|\mathbf{S})$ using $p(\mathbf{Z}|\mathbf{X})$, $p(\mathbf{X}|\mathbf{S})$ is transformed as

$$p(\mathbf{X}|\mathbf{S}) = \sum_{\mathbf{Z}} p(\mathbf{X}, \mathbf{Z}|\mathbf{S}) = \sum_{\mathbf{Z}} p(\mathbf{X}|\mathbf{Z}, \mathbf{S}) p(\mathbf{Z}|\mathbf{S}). \qquad (8)$$

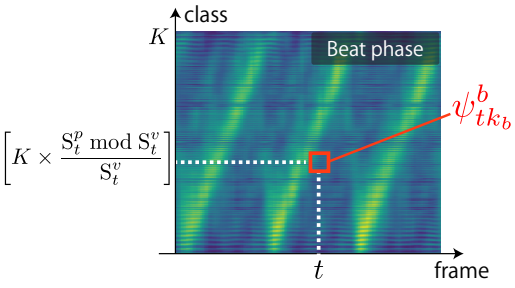

Figure 5. Observation probabilities $p(\mathbf{X}_t|\mathbf{S}_t)$ are represented as product of $\psi^b_{tk_b}$ and $\psi^d_{tk_d}$.

Let $\mathbf{Z_S}$ be the beat and downbeat series $\mathbf{Z}$ corresponding to the given latent state $\mathbf{S}$. Because $\mathbf{Z_S}$ is uniquely determined by $\mathbf{S}$, $p(\mathbf{X}|\mathbf{S})$ can be calculated as follows:

$$p(\mathbf{Z}|\mathbf{S}) = \delta_{\mathbf{Z},\mathbf{Z_S}}, \tag{9}$$

$$p(\mathbf{X}|\mathbf{S}) = \sum_{\mathbf{Z}} p(\mathbf{X}|\mathbf{Z},\mathbf{S})\delta_{\mathbf{Z},\mathbf{Z_S}} = p(\mathbf{X}|\mathbf{Z_S}). \tag{10}$$

Because $p(\mathbf{Z_S})$ is a uniform distribution and the term $p(\mathbf{X})$ is negligible, we get

$$p(\mathbf{X}|\mathbf{Z_S}) = \frac{p(\mathbf{Z_S}|\mathbf{X})p(\mathbf{X})}{p(\mathbf{Z_S})} \propto p(\mathbf{Z_S}|\mathbf{X}). \tag{11}$$

Finally, $p(\mathbf{X}|\mathbf{S})$ can be written using the estimated probabilities as follows:

$$p(\mathbf{X}|\mathbf{S}) \propto p(\mathbf{Z_S}|\mathbf{X}) = \prod_{t=1}^{T} \psi^b_{tk_b}\psi^d_{tk_d}, \tag{12}$$

$$k_b = \left[K \times \frac{S^p_t \bmod S^v_t}{S^v_t}\right], \tag{13}$$

$$k_d = \left[K \times \frac{S^p_t}{\mathrm{BPB} \cdot S^v_t}\right] = \left[K \times \frac{S^p_t}{N_t}\right]. \tag{14}$$

3.3 Tempo Estimation

Global tempo is computed from the beat phases estimated by the DNN using the DFT. The beat phase series is firstly converted into a sinusoidal curve $\mathbf{Y} \triangleq \{y_t\}_{t=1}^T$ as follows:

$$y_t = \sin\left(\frac{2\pi}{K} \times \arg\max_{1\le k\le K} \boldsymbol{\psi}^b_t\right). \tag{15}$$

The DFT is applied to the series $\mathbf{Y}$ and we can calculate the global tempo V as follows:

$$V = \frac{60\,\omega_{\max} \times \mathrm{fps}}{2\pi}[\mathrm{beats/60[s]}], \tag{16}$$

where $\omega_{\max}[\mathrm{rad/frame}]$ denotes the angular velocity with the largest absolute value of the Fourier coefficient in the DFT result. We can thus compute the most plausible tempo from a noisy beat phase sequence estimated by the DNN.

3.4 Joint Beat and Downbeat Estimation

We compare three architectures for joint estimation of beat and downbeat phases (Fig. 6). In the first architecture, we add another *Decoder* for downbeat estimation to the DNN described in Section 3.1. The components of the added decoder are the same as those used for beat estimation. The

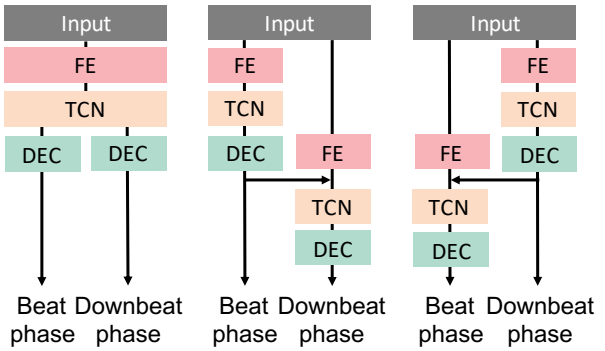

Figure 6. Three network architectures for joint beat and downbeat estimation. FE, TCN, and DEC correspond to "Feature embedding", "TCN layer×11", and "Decoder" in Fig. 2, respectively.

beat and downbeat phases are estimated in parallel. In the second architecture, the DNN described in Section 3.1 estimates beat or downbeat phases and then another DNN with the same architecture estimates the other, where the output of the former DNN in addition to the acoustic features are fed to the latter DNN. The third architecture estimates the beat and downbeat phases in the reverse order of the second architecture.

In the second architecture, when the output of the former DNN is fed to the latter DNN, the output probability series $\boldsymbol{\psi}^*$ is converted into the phase series $\hat{\mathbf{Z}}^* \triangleq \{\hat{z}^*_t\}_{t=1}^T$ as follows:

$$\hat{z_t}^* = \frac{2\pi}{K} \times \boldsymbol{a}^T \text{Gumbel-softmax}(\boldsymbol{\psi}^*_t), \tag{17}$$

where $\boldsymbol{a} = [1,\ldots,K]^T$ is a K-dimensional index vector and Gumbel-softmax$(\boldsymbol{\psi}^*_t)$ is a differentiable function that samples a random variable $\hat{z}^*_t$ that follows a discrete distribution $\boldsymbol{\psi}^*_t$. We concatenate the phase sequence $\hat{\mathbf{Z}}^*$ with the 20-dimensional vector obtained from the *Feature embedding* of the second DNN and input the 21-dimensional vector into the main *TCN layer* of the second DNN. Calculating the phase series in a differentiable state enables us to back-propagate the two DNNs at the same time.

4. EVALUATION

This section reports experiments conducted for validating the effectiveness of the proposed phase-aware DNN training and comparing the performances of the three multi-task learning architectures.

4.1 Experimental Conditions

To increase the amount of training data with various tempos, each song was pitch-shifted by -12, -6, $+6$, and $+12$ semitones and time-stretched by min_rate, (min_rate + 1) / 2, (max_rate + 1) / 2, and max_rate times, where min_rate and max_rate are given by

$$\text{min\_rate} = \frac{\mathrm{BPM_{min}}}{\mathrm{bpm}}, \quad \text{max\_rate} = \frac{\mathrm{BPM_{max}}}{\mathrm{bpm}}, \tag{18}$$

Method	F-measure	CMLt	AMLt
	RWC		
Baseline	0.835	0.717	0.85
Proposed	**0.846**	**0.765**	**0.861**
	Beatles		
Baseline	0.798	0.69	0.777
Proposed	**0.806**	**0.729**	**0.798**
	SMC		
Baseline	**0.502**	0.214	**0.424**
Proposed	0.428	**0.312**	0.39
	RWC		
Baseline	0.87	0.745	**0.909**
Proposed	**0.883**	**0.787**	0.901
	Beatles		
Baseline	**0.847**	**0.763**	**0.866**
Proposed	0.834	0.748	0.819
	SMC		
Baseline	**0.569**	**0.466**	**0.621**
Proposed	0.538	0.422	0.565

Table 1. The performances of beat estimation. The upper half of the table is the result using peak-picking and the lower half using the DBN as a post-processing step.

where bpm is the original tempo and $\text{BPM}_{\text{max}} = 215$ and $\text{BPM}_{\text{min}} = 55$ were used. The log-magnitude spectrogram was obtained by short-time Fourier transform (STFT) with a window size of 2048 and a hop length of 441 (100 frames per second) and then transformed into the log-frequency scale consisting of 81 bins from 30 to 17,000 Hz.

The DNN was trained with Adam optimizer [24]. The batch size was 1. The learning rate was initialized to 1×10^{-3} and then halved gradually if the validation loss did not improve for 15 epochs. The training was terminated when the validation loss did not improve for 50 epochs or when 200 epochs were finished. To prevent gradient explosion, gradients greater than 0.5 were clipped. The kernel size of the dilated convolutions was set to 5 and the dropout rate was set to 0.1. The phase value was quantized into $K = 360$ classes. Other parameters are shown in Fig. 3. In the DBN-based post-processing, the maximum and minimum BPMs were the same as those in the data augmentation, and the λ was set to 100.

We separately conducted eight-fold cross validations on the RWC Popular Music dataset [25] and the Beatles dataset [23]. Although our main target is popular music, we also used the SMC dataset [26], which contains various genres such as classical, blues, and film music. As in [7], we used the F-measure, CMLt, and AMLt as evaluation metrics for beat and downbeat estimation, and Accuracy 1 and Accuracy 2 for tempo estimation. F-measure has a tolerance window of $\pm$ 70 ms around the ground-truth beats. CMLt considers a beat to be correct if its tempo and phase are within a 17.5% tolerance of those of the ground-truth beat. In addition to CMLt, AMLt allows a series of double/half and triple/third tempo of the annotated beats. Accuracy 1 considers an estimated tempo correct with a tolerance of 4% of the correct tempo, and Accuracy 2 considers correct

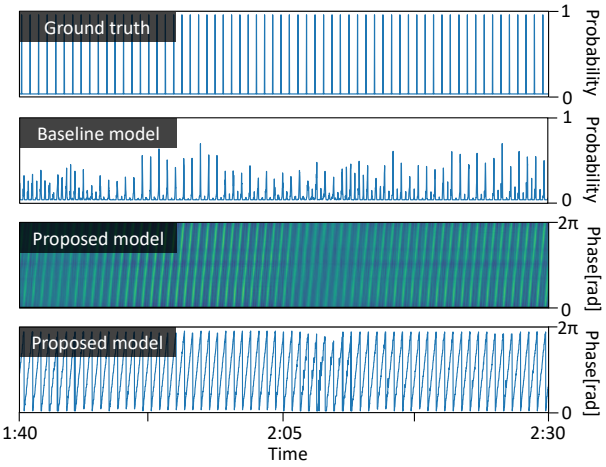

Figure 7. Estimation results for "Dear Prudence" by The Beatles. From top to bottom, the ground truth, the beat presence probability series estimated by the baseline method, the probability of each phase at each time estimated by the proposed method, and the phase series with the highest probability at each time in the third panel.

if the estimated tempo satisfies the Accuracy 1 also with tempo in double/half and triple/third target tempo.

First, we compared methods that estimate only beats. The proposed method estimates beat phases, whereas the baseline method estimates beat presence probabilities. The decoder of the baseline method was modified so that it consists of dropout, dense, and sigmoid layers as in [7–9]. We tested both a naive peak-picking method and the periodicity-aware DBN in a post-processing step that detects beat times. The peak-picking algorithm first identifies all the maxima and then selects the peaks with intervals greater than a specified horizontal distance in the order of increasing magnitude [2] . We used 40 frames (400ms) as the distance value.

Next, we compared the three multitask learning architectures that jointly estimate beat and downbeat times followed by tempo estimation. For comparison, we implemented the existing method [7] by adding another decoder for downbeat estimation that is equivalent to the decoder for beat estimation in the baseline method, and adding the decoder for tempo estimation described in [7].

4.2 Experimental Results

Table 1 shows the performances of the proposed and baseline methods that estimate only beat times on the RWC, Beatles, and SMC datasets. When the basic peak-picking method was used for post-processing, the proposed method outperformed the baseline method, especially in terms of the CMLt by a large margin. This indicates that the proposed method better captures the periodic nature of metrical structure, resulting in the better regularity of the estimated beat times. We found that the proposed method achieved a better CMLt for not only popular music (RWC

[2] We use the peak-picking function specified here: https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.find_peaks.html

	Beat			*Downbeat*			*Tempo*	
	F-measure	CMLt	AMLt	F-measure	CMLt	AMLt	Accuracy 1	Accuracy 2
				RWC				
beat-and-downbeat	0.907	0.817	0.907	0.878	0.822	0.887	0.861	**1.00**
beat-to-downbeat	0.884	0.783	0.902	0.854	0.802	0.880	0.850	0.990
downbeat-to-beat	**0.920**	**0.845**	0.914	0.884	0.843	0.890	**0.900**	0.990
Böck *et al.* [7]	0.914	0.83	**0.952**	**0.902**	**0.850**	**0.941**	0.853	0.980
				Beatles				
beat-and-downbeat	0.823	0.722	0.786	0.753	0.683	0.748	0.872	0.955
beat-to-downbeat	0.832	0.738	0.819	0.774	0.703	0.778	0.861	0.961
downbeat-to-beat	0.825	0.740	0.800	0.767	0.708	0.767	**0.883**	0.966
Böck *et al.* [7]	**0.862**	**0.779**	**0.895**	**0.825**	**0.767**	**0.871**	0.860	**0.967**

Table 2. The performances of joint beat and downbeat estimation. "beat-and-downbeat" denotes the architecture that simultaneously estimates beat and downbeat, "beat-to-downbeat" denotes the architecture that first estimates beats and subsequently estimates downbeats, and "downbeat-to-beat" denotes the reverse version of the "beat-to-downebat".

and Beatles dataset) but also various music genres (SMC dataset). When the DBN was used for post-processing, in contrast, the baseline method worked best in almost all cases except for the F-measure and CMLt on the RWC dataset. Further refinement of the DBN suitable for a sequence of phases should be investigated in the future.

Fig. 7 shows an example of the estimation results obtained by the baseline and proposed methods. While the baseline method showed high beat probabilities even at non-beat times, the proposed method estimated high probabilities at the target phases. Thus, our method has the potential to continuously estimate the phase with a constant angular velocity, This is in line with the high accuracy of CMLt in peak-picking results. However, as can be seen from the third panel in Fig. 7, the phase probabilities in the proposed method tend to be blurred in difficult segments to detect the periodicity. The baseline method, in such segments, showed high probabilities in aperiodic locations instead of blurring the beat probabilities. This difference in the output behavior in the difficult section is considered to have an effect on the DBN.

Table 2 shows the comparison of the methods used for simultaneously estimating beat, downbeat, and tempo. In the three architectures of the proposed method, "downbeat-to-beat" worked best for the RWC dataset, whereas "beat-to-downbeat" worked well for the Beatles dataset. Because downbeat estimation can be performed accurately for the RWC dataset compared with the Beatles dataset, the beat detection of "downbeat-to-beat" is considered to have the best accuracy by leveraging the downbeat estimation. By contrast, in the Beatles dataset, the estimated beat phases are considered to help improve the downbeat estimation because "beat-to-downbeat" had higher accuracy. In both datasets, "beat-and-downbeat" did not show the highest accuracy in the beat and downbeat evaluation. We consider that this is because "beat-to-downbeat" or "downbeat-to-beat" can use additional information for training and estimation. For example, in the case of "beat-to-downbeat", the latter DNN which estimates downbeats can utilize the result of beat estimation, which is expected to improve the downbeat estimation, and the parameters of the former DNN are optimized to output the better downbeat estimation, which is expected to improve the beat estimation as well. In the results of tempo estimation, "downbeat-to-beat" showed the best accuracy in Accuracy 1. This is considered to have a relationship with the better accuracy of CMLt in our method when applying the peak-picking method since our tempo estimation method relies on the estimated phase series.

5. CONCLUSION

We proposed a phase-aware beat, downbeat, and tempo estimation method. More specifically, we trained a DNN to estimate a phase at each frame instead of a beat presence probability and calculate beat and downbeat times and tempo on the basis of the estimated phase. The experimental results showed that the proposed method could estimate more periodic beats than the conventional method that depends on a post-processing step.

For future work, we plan to utilize tempo to estimate more periodic beat times. Considering that beat and downbeat times are closely related to the other components used in MIR (*e.g.* drum scores), it would be beneficial to train an end-to-end model that directly estimates beat times and other components from music signals in the framework of multi-task learning.

6. ACKNOWLEDGEMENT

This work is funded by JST PRESTO No. JPMJPR20CB and JSPS KAKENHI Nos. 19H04137 and 20K21813.

7. REFERENCES

[1] R. Nishikimi, E. Nakamura, M. Goto, K. Itoyama, and K. Yoshii, "Bayesian singing transcription based on a hierarchical generative model of keys, musical notes, and F0 trajectories," in *IEEE/ACM Transactions on Audio, Speech, and Language Processing (TASLP)*, 2020, pp. 1678–1691.

[2] R. Ishizuka, R. Nishikimi, and K. Yoshii, "Global structure-aware drum transcription based on self-attention mechanisms," in *arXiv*, 2021.

[3] G. Shibata, R. Nishikimi, and K. Yoshii, "Music structure analysis based on an LSTM-HSMM hybrid model," in *International Society for Music Information Retrieval Conference (ISMIR)*, 2020, pp. 15–22.

[4] G. Peeters, "Beat-marker location using a probabilistic framework and linear discriminant analysis," in *International Conference on Digital Audio Effects (DAFx)*, 2009.

[5] D. P. Ellis, "Beat tracking by dynamic programming," in *Journal of New Music Research*, 2007, pp. 51–60.

[6] M. E. Davies and M. D. Plumbley, "Context-dependent beat tracking of musical audio," in *IEEE/ACM Transactions on Audio, Speech, and Language Processing (TASLP)*, 2007, pp. 1009–1020.

[7] S. Böck and M. E. Davies, "Deconstruct, analyse, reconstruct: How to improve tempo, beat, and downbeat estimation," in *International Society for Music Information Retrieval Conference (ISMIR)*, 2020, pp. 574–582.

[8] S. Böck, M. E. Davies, and P. Knees, "Multi-task learning of tempo and beat: Learning one to improve the other," in *International Society for Music Information Retrieval Conference (ISMIR)*, 2019, pp. 486–493.

[9] E. MatthewDavies and S. Böck, "Temporal convolutional networks for musical audio beat tracking," in *European Signal Processing Conference (EUSIPCO)*, 2019, pp. 1–5.

[10] S. Böck, F. Krebs, and G. Widmer, "Joint beat and downbeat tracking with recurrent neural networks," in *International Society for Music Information Retrieval Conference (ISMIR)*, 2016, pp. 255–261.

[11] ——, "A multi-model approach to beat tracking considering heterogeneous music styles," in *International Society for Music Information Retrieval Conference (ISMIR)*, 2014, pp. 603–608.

[12] W. Nick, T. C. Ali, and J. G. Simon, "Bayesian modelling of temporal structure in musical audio," in *International Society for Music Information Retrieval Conference (ISMIR)*, 2006, pp. 29–34.

[13] F. Krebs, S. Böck, and G. Widmer, "An efficient state-space model for joint tempo and meter tracking," in *International Society for Music Information Retrieval Conference (ISMIR)*, 2015, pp. 72–78.

[14] S. Böck and M. Schedl, "Enhanced beat tracking with context-aware neural network," in *International Conference on Digital Audio Effects (DAFx)*, 2011, pp. 135–139.

[15] S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," in *arXiv preprint arXiv:1803.01271*, 2018.

[16] A. v. d. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," in *arXiv preprint arXiv:1609.03499*, 2016.

[17] G. Percival and G. Tzanetakis, "Streamlined tempo estimation based on autocorrelation and cross-correlation with pulses," in *IEEE/ACM Transactions on Audio, Speech, and Language Processing (TASLP)*, 2014, pp. 1765–1776.

[18] S. Böck, F. Krebs, and G. Widmer, "Accurate tempo estimation based on recurrent neural networks and resonating comb filters," in *International Society for Music Information Retrieval Conference (ISMIR)*, 2015, pp. 625–631.

[19] F.-H. F. Wu, T.-C. Lee, J.-S. R. Jang, K. K. Chang, C.-H. Lu, and W.-N. Wang, "A two-fold dynamic programming approach to beat tracking for audio music with time-varying tempo," in *International Society for Music Information Retrieval Conference (ISMIR)*, 2011, pp. 191–196.

[20] H. Schreiber and M. Müller, "A single-step approach to musical tempo estimation using a convolutional neural network," in *International Society for Music Information Retrieval Conference (ISMIR)*, 2018, pp. 98–105.

[21] H. Foroughmand and G. Peeters, "Deep-rhythm for tempo estimation and rhythm pattern recognition," in *International Society for Music Information Retrieval Conference (ISMIR)*, 2019.

[22] M. Goto, "An audio-based real-time beat tracking system for music with or without drum-sounds," in *Journal of New Music Research*, 2001, pp. 159–171.

[23] M. E. Davies, N. Degara, and M. D. Plumbley, "Evaluation methods for musical audio beat tracking algorithms," in *Queen Mary University of London, Centre for Digital Music, Tech. Rep. C4DM-TR-09-06*, 2009.

[24] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *International Conference on Learning Representations (ICLR)*, 2015.

[25] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka, "RWC music database: Popular, classical and jazz music databases," in *International Society for Music Information Retrieval Conference (ISMIR)*, 2002, pp. 287–288.

[26] A. Holzapfel, M. E. P. Davies, J. R. Zapata, J. L. Oliveira, and F. Gouyon, "Selective sampling for beat tracking evaluation," in *IEEE/ACM Transactions on Audio, Speech, and Language Processing (TASLP)*, 2012, pp. 2539–2548.