2023 Asia Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)

2023 Asia Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC) | 979-8-3503-0067-3/23/$31.00 ©2023 IEEE | DOI: 10.1109/APSIPAASC58517.2023.10317515

CTC2: End-to-End Drum Transcription Based on Connectionist Temporal Classification With Constant Tempo Constraint

Daichi Kamakura, Eita Nakamura, and Kazuyoshi Yoshii
Graduate School of Informatics, Kyoto University, Japan
E-mail: {kamakura, enakamura}@sap.ist.i.kyoto-u.ac.jp, yoshii@i.kyoto-u.ac.jp

***Abstract*—This paper describes end-to-end automatic drum transcription for directly estimating a drum score from an audio signal of popular music using non-aligned paired data. We aim to convert a sequence of frame-level acoustic features into a sequence of tatum-level score fragments (three-dimensional multi-hot vectors) representing the presence or absence of the onsets of the bass and snare drums and the hi-hats. The main challenge of this task lies in estimating the correct number of inactive tatums having no onset between active tatums. One may use the connectionist temporal classification (CTC) for end-to-end training of a deep neural network (DNN) that infers a frame-level state sequence (alignment path) including the special "blank" states representing the tatum boundaries. At run-time, a drum score is obtained by annexing repeated states and removing all blank states from the most likely frame-level state sequence. This approach, however, tends to yield a shortened drum score in which repeated inactive tatums are annexed mistakenly because the blank state (tatum boundary) cannot be distinguished acoustically from the inactive state (onset absence) at the frame level. In this paper, we propose a sophisticated version of the CTC with constant tempo constraint, CTC2 in short, that encourages each tatum to be aligned with almost the same number of frames. Although the loss function can be computed efficiently as in the basic CTC, the backpropagation over the huge computation graph made through the forward algorithm is computationally prohibitive. To solve this problem, we propose to perform the backpropagation with only an alignment path stochastically drawn with Gibbs sampling. The experiment showed that the proposed method worked well as expected.**

I. Introduction

Automatic drum transcription (ADT) is a fundamental task that aims to estimate a drum score (MusicXML format) from a music signal. Although ADT plays a key role in computational music understanding, it has been tackled only partially; most studies aim to estimate a piano roll (MIDI format) by detecting the onset times of drums in seconds [1], [2]. To estimate a drum score, for example, one may estimate the beat and downbeat times in advance [3], [4] and then perform quantized onset detection [5]. Instead, one may take a more sophisticated approach in the same way as the state-of-the-art automatic piano transcription [6] based on the combination of audio-to-MIDI transcription [7] and MIDI-to-score transcription (rhythm transcription) [8]. Such a cascading approach, however, suffers from the error propagation problem.

Inspired by the great success of end-to-end automatic speech recognition (ASR), end-to-end automatic music transcription (AMT) has recently been investigated [9]–[13]. This approach can circumvent the error propagation problem and make effective use of non-aligned pairs of music signals and the corresponding scores as training data. Considering the monotonic nature of the audio-to-score mapping, we focus on the connectionist temporal classification (CTC) [14] (cf. [9]) because it is expected to work stably with a limited amount of training data, compared with the encoder-decoder model (cf. [10]) and that with the attention mechanism [15] (cf. [11]–[13]).

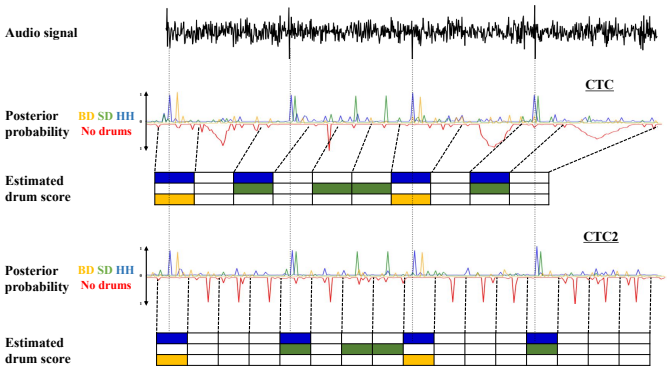

Fig. 1. The comparison of the basic CTC and the proposed CTC with constant tempo constraint (CTC2). The repeated inactive tatums having no drum onsets can be estimated correctly by encouraging the frames to be segmented into the tatums with almost the same length.

End-to-end AMT methods can also be characterized by their score representations: how to define symbols constituting the output sequence. The representation affects the difficulty unique to AMT in estimating the temporal information about note values (quantized durations) and metrical structures. The basic way is to convert a sequence of frame-level acoustic features into a sequence of *notation-level score components* (e.g., notes and bars) [12]. The note values (temporal attribute), however, are considerably harder to estimate than the note pitches (instantaneous attribute) with the basic input-output alignment mechanism. Another way is to estimate a sequence of *tatum-level score fragments* [11], where the tatum is a basic time unit on the score at the sub-beat level (typically at the sixteenth-note level). Instead of directly estimating the note values, this approach aims to estimate the presence or absence of the note onsets (instantaneous attribute) at the tatum level.

In this paper we tackle CTC-based ADT that aims to estimate a sequence of tatum-level score fragments (Fig. 1). Each fragment (tatum) is represented as a three-dimensional multi-hot vector indicating the presence or absence of the onsets of the bass and snare drums and the hi-hats (eight states in total). The main challenge unique to this task is to estimate the correct number of inactive tatums (a state corresponding to the all-zero vector) between active tatums (the other seven states). If the basic CTC is used for end-to-end training, the state of each tatum of the ground-truth sequence is associated with frames of the input sequence whose acoustic features are particularly relevant to the state through forced alignment. Such frames are typically only at the beginning of the tatum; the special blank state (denoted by "_") is associated with the other acoustically irrelevant frames. At run-time, however, successive inactive tatums are annexed mistakenly because the blank state (tatum boundary) and the inactive state (onset absence) are hard to distinguish acoustically at the frame level.

To solve this problem, we propose a sophisticated version of the CTC with constant tempo constraint, CTC2 in short, based on the reasonable assumption that the tempo is usually kept constant throughout the song in popular music (Fig. 1). In the basic CTC, the frame-to-tatum alignment is obtained through unconstrained time stretch because only the order of output symbols (states) matters and their durations are ignored, as in the inference of the latent state sequence of the hidden Markov model (HMM). Inspired by the hidden semi-Markov model (HSMM) with explicit duration modeling [16], we instead encourage the tatum boundaries (transitions between the main eight states and the blank state) to occur regularly with a constant interval, where adjacent tatums favor to have the same duration [3], [17]. This makes successive inactive frames split into an appropriate number of successive inactive tatums at the frames associated with the blank state.

We also propose an efficient training method that works with the CTC and its variants. The objective function of the CTC2 to be maximized is given by the sum of the posterior probabilities of all possible alignment paths as in the CTC. Although it can thus be computed efficiently with dynamic programming (forward algorithm) in the analogy of the HMM to the HSMM, the backpropagation over the huge computation graph is computationally prohibitive. We thus perform the backpropagation along only an alignment path stochastically drawn according to the posterior probability with Gibbs sampling. Technically, we run the forward filtering-backward sampling in the evaluation mode to draw a path and then make a compact computation graph along the path in the training mode.

II. Related Work

This section reviews related work on ADT and that on end-to-end AMT.

A. Automatic Drum Transcription

Deep neural networks (DNNs) have intensively been used for ADT [1], [2], [5], [18]–[22]. In general, the spectrogram of an audio signal is used as input, and the annotated drum onset times are used as training data. For feature extraction, convolutional neural networks (CNNs) have commonly been used. Regularization based on prior information about drum scores [5] and refined network architectures [23] have also been attempted, resulting in improved performance of transcription. For converting DNN outputs into musical scores, we use beat information for quantization, and it has been reported that using temporal convolutional networks (TCNs) [24], [25] for beat tracking is effective. However, one problem with these methods is that they require as training data a sufficient amount of music spectrograms with precise annotations of drum onset times. To address this issue, the use of synthetic datasets [26], data augmentation [19], and unsupervised learning [27] have been proposed. Nevertheless, these approaches still face challenges in terms of robustness against real performances and the ability to handle variations in drum sounds.

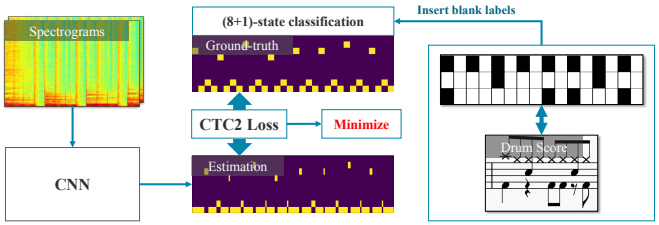

Fig. 2. The proposed ADT method based on the CTC2.

B. End-to-End Automatic Music Transcription

For end-to-end AMT that aims to directly estimate a musical score (a sequence of musical notations on the score), the attention mechanism [12] and the CTC [28], [29] have been investigated. In singing transcription based on the attention mechanism, the tempo consistency was taken into account [12], where the frame-to-tatum alignment is encouraged to be monotonic and regular by imposing a regularization term on the attention matrix. Such regularization, however, was found to prevent the initial progress of the training.

III. Proposed Method

This section describes the audio-to-score ADT method based on the CTC2-based end-to-end training (Fig. 2).

A. Problem Specification

Our goal is to estimate a drum score from a sequence of the stereo power spectra of a target musical piece, $\mathbf{X} \triangleq \{\mathbf{x}_t\}_{t=1}^{T}$ ($\mathbf{x}_t \in \mathbb{R}^{2F}$), where F is the number of frequency bins and T is the number of frames. We aim to estimate the presence or absence of the onsets of I drum instruments at the tatum level. In this study, we focus on the three common drum instruments: bass drum (BD), snare drum (SD), and hi-hats (HH) ($I = 3$). Let $y_n \in \{1, \ldots, K\}$ be the state of tatum n, which is defined as $y_n = 1 + 2^0\langle \mathtt{BD} \rangle_n + 2^1\langle \mathtt{SD} \rangle_n + 2^2\langle \mathtt{HH} \rangle_n$, where $K = 2^I = 8$, and $\langle \mathtt{DR} \rangle_n$ is a binary value indicating the presence or absence of an onset of DR at the tatum. Let $\mathbf{Y} \triangleq \{y_n\}_{n=1}^{N}$, where N is the number of tatums. The tatum is defined as the one-fourth of the beat, i.e., the tatum and beat correspond to the durations of the sixteenth and quarter notes, respectively.

B. Training and Inference

We describe the basic flow of end-to-end ADT. In addition to the main states indexed by $\{1, 2, \dots, K\}$ (Section III-A), we introduce the special blank state indexed by 0. Let $\boldsymbol{\pi} \triangleq \{\pi_t\}_{t=1}^{T}$ be a redundant state sequence, where $\pi_t \in \{0, \dots, K\}$ represents the state of frame t. We use a DNN with parameters θ that outputs the posterior probabilities of the $(K+1)$ states at the frame level, denoted by $\boldsymbol{\phi} \triangleq \{\phi_{k,t}\}_{k=0,t=1}^{K,T}$, where $\phi_{k,t}$ represents the probability of state k at frame t.

1) Training: Given a non-aligned pair of $\mathbf{X}$ and $\mathbf{Y}$ as training data, we train the DNN. Let $\mathcal{B}(\boldsymbol{\pi})$ be a one-to-one reducer that returns $\mathbf{Y}$ by annexing repeated states and removing all blank states from $\boldsymbol{\pi}$, e.g., $\mathcal{B}(01110112) = \mathcal{B}(01100122) = 112$. Let $\mathcal{B}^{-1}(\mathbf{Y})$ be a one-to-many expander that returns a set of all possible redundant state sequences that reduce to $\mathbf{Y}$, e.g., $\mathcal{B}^{-1}(112) = \{01110112, 1100122, \dots\}$. The DNN parameters θ are optimized such that the following posterior probability is maximized:

$$\begin{aligned}\mathcal{L} &= \log p(\mathbf{Y}|\mathbf{X}, \theta) \\ &= \log \sum_{\boldsymbol{\pi} \in \mathcal{B}^{-1}(\mathbf{Y})} p(\boldsymbol{\pi}|\mathbf{X}, \theta). \end{aligned} \tag{1}$$

2) Inference: Given $\mathbf{X}$, we estimate $\mathbf{Y}$ using a trained DNN with the parameters θ. The redundant state sequence $\boldsymbol{\pi}$ is obtained by selecting the most likely state at each frame:

$$\pi_t = \operatorname*{argmax}_k \phi_{k,t}. \tag{2}$$

The final output $\mathbf{Y}$ is given by $\mathbf{Y} = \mathcal{B}(\boldsymbol{\pi})$.

C. Conventional CTC

As a baseline, we briefly explain the training method based on the basic CTC. The posterior probability of an alignment path $\boldsymbol{\pi}$ in (1) is given by the product of the frame-wise posterior probabilities of the states as follows:

$$p(\boldsymbol{\pi}|\mathbf{X}, \theta) = \prod_{t=1}^{T} \phi_{\pi_t, t}. \tag{3}$$

The loss function $\mathcal{L}$ or $p(\mathbf{Y}|\mathbf{X}, \theta)$ in (1) can be computed efficiently with a dynamic programming technique called the forward algorithm as in the HMM. Let $\mathbf{Y}' \triangleq \{y'_s\}_{s=0}^{S}$ be an expanded state sequence obtained by inserting the $N-1$ blank states between the N states of $\mathbf{Y}$ and pushing the blank states to the front and back of $\mathbf{Y}$, i.e., $S = 2N+1$, $y'_{2n-1} = y_n$, and $y'_s = y_{\lfloor \frac{s}{2}+1 \rfloor}$. Let $\alpha_t(s)$ be the forward probability obtained by accumulating the posterior probabilities of all possible paths that align x_t with y'_s:

$$\alpha_t(s) = \sum_{\pi_{1:t} \in \mathcal{B}^{-1}(y_{1:\lfloor \frac{s}{2} \rfloor})} \prod_{\tau=1}^{t} \phi_{\pi_\tau, \tau}, \tag{4}$$

where $a_{1:t} \triangleq \{a_1, \dots, a_t\}$. Since an alignment path $\boldsymbol{\pi}$ is valid if the last frame x_T is aligned with the last non-blank symbol $y'_{S-1} = y_n$ or the last blank symbol y'_S, we have

$$p(\mathbf{Y}|\mathbf{X}, \theta) = \alpha_T(S-1) + \alpha_T(S). \tag{5}$$

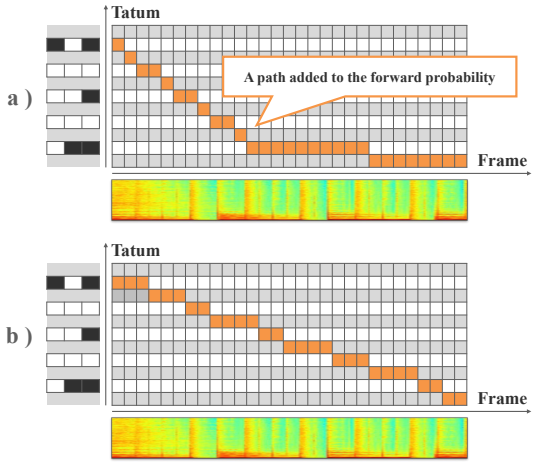

Fig. 3. Examples of the alignment path $\boldsymbol{\pi}$ estimated by a) the CTC- and b) CTC2-based training methods.

The forward probabilities can be computed recursively with the time complexity of $O(TS)$. First, we set the initial values:

$$\alpha_1(1) = \phi_{0,1}, \tag{6}$$

$$\alpha_1(2) = \phi_{y_1,1}, \tag{7}$$

$$\alpha_1(s) = 0, \forall s > 2. \tag{8}$$

We then use the following recursive formulas:

$$\alpha_t(s) = \begin{cases} \text{if } y'_s = \text{blank or } y'_{s-2} \\ \quad [\alpha_{t-1}(s) + \alpha_{t-1}(s-1)]\, \phi_{\pi_s, t}, \\ \text{else} \\ \quad [\alpha_{t-1}(s) + \alpha_{t-1}(s-1) + \alpha_{t-1}(s-2)]\, \phi_{\pi_s, t}. \end{cases} \tag{9}$$

In the inference stage, however, the optimal alignment path $\boldsymbol{\pi}$ estimated by (2) tends to deviate considerably from the diagonal line on the alignment map between the input sequence $\mathbf{X}$ and the estimated sequence $\mathbf{Y}'$ (Fig. 3-a). The durations of the N symbols in $\mathbf{Y} = \mathcal{B}(\boldsymbol{\pi})$ thus have a large variation, meaning that the tempo is allowed to frequently change at the tatum level in a musically-unnatural manner.

D. Training with Proposed CTC2

We then explain the training method based on the proposed CTC2 for explicit duration modeling with the constant tempo constraint (Fig. 4). Let $\mathbf{D} = \{d_n\}_{n=1}^{N}$ $(D_{\min} \le d_n \le D_{\max})$ denote the durations of the N symbols of $\mathbf{Y}$, where $D_{\min}$ and $D_{\max}$ are the minimum and maximum durations to be considered. When the frame shift is 10 [ms], the local tempo at tatum n is $1500/d_n$ [bpm]. Let $\mathcal{B}(\boldsymbol{\pi})$ be redefined as a one-to-one reducer that returns a pair $(\mathbf{Y}, \mathbf{D})$, e.g., $\mathcal{B}(01110112) = (112, 421)$, $\mathcal{B}(01100122) = (112, 412)$. Let $\mathcal{B}^{-1}(\mathbf{Y}, \mathbf{D})$ be redefined as a one-to-many expander that returns all possible sequences that reduce to $(\mathbf{Y}, \mathbf{D})$, e.g., $\mathcal{B}^{-1}(112, 412) = \{01110112, 1100122, \dots\}$. Note that even if both $\mathbf{Y}$ and $\mathbf{D}$ are given, $\mathcal{B}^{-1}(\mathbf{Y})$ may return multiple paths.

The posterior probability of $\mathbf{Y}$ in (1) is given by

$$p(\mathbf{Y}|\mathbf{X}, \theta) = \sum_{\mathbf{D}} p(\mathbf{Y}|\mathbf{X}, \theta, \mathbf{D}) p(\mathbf{D}), \tag{10}$$

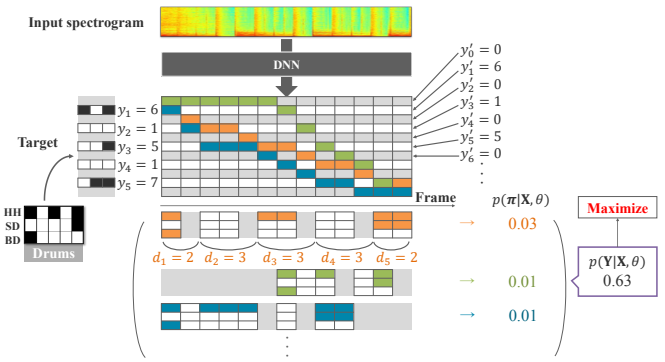

Fig. 4. The forward algorithm in the proposed CTC2-based training with explicit duration modeling.

where $p(\mathbf{Y}|\mathbf{X},\theta,\mathbf{D})$ is the posterior probability of $\mathbf{Y}$ conditioned by $\mathbf{D}$ and $p(\mathbf{D})$ is the generative probability of $\mathbf{D}$ given by a Markov model as follows:

$$p(\mathbf{D}) = p(d_1)\prod_{n=2}^{N} p(d_n|d_{n-1}), \tag{11}$$

$$p(d_1) \propto 1, \tag{12}$$

$$p(d_n|d_{n-1}) \propto \exp\left(-\lambda\left|\frac{d_n}{d_{n-1}} - 1\right|\right). \tag{13}$$

The loss function $\mathcal{L}$ or $p(\mathbf{Y}|\mathbf{X},\theta)$ in (1) can be computed efficiently as in the HSMM. Let $\alpha_t(s,d,c)$ be the forward probability obtained by accumulating the posterior probabilities of all possible paths that align x_t with y'_s:

$$\alpha_t(s,d,c) = \sum_{\pi_{1:t}\in\mathcal{B}^{-1}(y_{1:\lfloor\frac{s}{2}\rfloor},d_{1:\lfloor\frac{s}{2}\rfloor})} \prod_{\tau=1}^{t} \phi_{\pi_\tau,\tau}\cdot p(d_{1:\lfloor\frac{s}{2}\rfloor}), \tag{14}$$

$$p(d_{1:\lfloor\frac{s}{2}\rfloor}) = p(d_1)\prod_{n=2}^{\lfloor\frac{s}{2}\rfloor} p(d_n|d_{n-1}), \tag{15}$$

where d represents the duration of the current main state $y_{\lfloor\frac{s}{2}\rfloor}$ and c is a counter variable that is set to d when a new state starts and is then decremented until the state ends. We have

$$p(\mathbf{Y}|\mathbf{X},\theta) = \sum_{s=S-1}^{S}\sum_{d=D_{\min}}^{D_{\max}}\sum_{c=1}^{d}\alpha_T(s,d,c). \tag{16}$$

Note that if $\lambda = 0$ in (13), i.e., the durations of the output symbols are allowed to take any values with the same probability, i.e., $p(\mathbf{D}) = 1$, the CTC2 reduces to the CTC, i.e., (14) and (16) reduce to (4) and (5), respectively.

The forward probabilities can be computed recursively with the time complexity of $O(TSD^2)$, where $D \triangleq D_{\max} - D_{\min} + 1$. First, we set the initial values:

$$\alpha_1(1,d,c) = \phi_{0,1}, \forall d,c, \tag{17}$$

$$\alpha_1(2,d,c) = \phi_{y_1,1}, \forall d,c, \tag{18}$$

$$\alpha_1(s,d,c) = 0, \forall s > 2. \tag{19}$$

We then use the following recursive formulas:

$$\alpha_t(s,d,c) = \begin{cases} \text{if } d = c \\ \quad \text{if } y'_s = \text{blank} \\ \quad\quad 0, \\ \quad \text{else if } y'_s = y'_{s-2} \\ \quad\quad \sum_{d'}\alpha_{t-1}(s-1,d',1)\cdot\phi_{y'_s,t}\cdot p(d|d'), \\ \quad \text{else} \\ \quad\quad \sum_{d'}\bar{\alpha}_t(s,d,c)\cdot\phi_{y'_s,t}\cdot p(d|d'), \\ \text{else if } d > c \\ \quad \text{if } y'_s = \text{blank} \\ \quad\quad \bar{\bar{\alpha}}_t(s,d,c)\cdot\phi_{y'_s,t}, \\ \quad \text{else} \\ \quad\quad \alpha_{t-1}(s,d',c+1)\cdot\phi_{y'_s,t}, \\ \text{else} \\ \quad 0, \end{cases} \tag{20}$$

$$\bar{\alpha}_t(s,d,c) = \alpha_{t-1}(s-1,d,1) + \alpha_{t-1}(s-2,d,1), \tag{21}$$

$$\bar{\bar{\alpha}}_t(s,d,c) = \alpha_{t-1}(s,d,c+1) + \alpha_{t-1}(s-1,d,c+1). \tag{22}$$

In the inference stage, the optimal alignment path $\boldsymbol{\pi}$ estimated by (2) tends to roughly follow the diagonal line on the alignment map between the input sequence $\mathbf{X}$ and the estimated sequence $\mathbf{Y}'$ (Fig. 3-b). The durations $\mathbf{D}$ of the N symbols of $\mathbf{Y}$, which are given by $(\mathbf{Y},\mathbf{D}) = \mathcal{B}(\boldsymbol{\pi})$, are thus kept almost constant. This enables the active and inactive states (tatums) to have similar durations.

The CTC2, however, considers several orders of magnitude larger number of possible paths than the CTC and thus makes the exact backpropagation prohibitive in practice. Note that the forward computation in the inference mode without constructing the computation graph can still be performed quickly.

E. Computationally-Efficient Training

We propose an efficient training method that can be applied to the CTC2 (and the CTC) at a minimum sacrifice of performance. Instead of backpropagating the error through all possible paths on the huge computation graph constructed by the forward algorithm, we focus on only the most likely path found by the Viterbi algorithm or a path randomly selected according to its posterior probability with the Gibbs sampling. To determine such a path, the forward and backward recursions can be executed faster in the inference mode. Since the Gibbs sampling stochastically generates likely paths not limited to the most likely one, it is expected to be more robust against local optima than the Viterbi algorithm. Let $\mathbf{Z} \triangleq \{z_t\}_{t=1}^{T}$ be a latent state sequence, where $z_t \triangleq (s_t,d_t,c_t)$. An alignment path $\boldsymbol{\pi}$ is uniquely determined by $\mathbf{Z}$.

1) Viterbi Algorithm: In the forward recursion, the forward probabilities $\{\alpha_t(z_t)\}_{t=1}^{T}$ are computed as described in Section III-D except that the sum operations over d' of (20) and the additions of (21) and (22) are replaced with the max operations, where the indices and terms that take the maximum values are memorized. In the backward recursion, the states of most likely $\mathbf{Z}$ are determined in the reverse order by backtracing the memorized path from z_T such that $\alpha_T(z_T)$ is maximized, where $s_T \in \{S-1,S\}$, $d_T \in \{D_{\min},\ldots,D_{\max}\}$, and $c_T \in \{1,\ldots,d_T\}$.

2) Gibbs Sampling: We use the forward filtering-backward sampling algorithm, which was originally proposed for sampling the latent sequence of an HMM or HSMM [30]. In the forward filtering, the forward probabilities are computed as described in Section III-D. In the backward sampling, the states of $\mathbf{Z}$ are sampled in the reverse order according to

$$p(z_T) \propto \alpha_T(z_T), \tag{23}$$

$$p(z_t|z_{t+1:T}) \propto \alpha_t(z_t) A_{z_t z_{t+1}} \ (t = \{T-1, \ldots, 1\}), \tag{24}$$

where $A_{z_t z_{t+1}}$ represents the transition probability from z_t to z_{t+1} obtained by normalizing the following matrix:

$$A'_{z_t z_{t+1}} = \begin{cases} p(d_t|d_{t+1}) & \text{if condition 1 holds,} \\ 1 & \text{if condition 2 holds,} \\ 0 & \text{otherwise,} \end{cases} \tag{25}$$

condition 1: $c_t = 1$ and ($y'_s =$ blank and $s_{t+1} = s_t + 1$ or $y'_s \neq$ blank and $y'_s \neq y'_{s+2}$ and $s_{t+1} = s_t + 2$),

condition 2: $d_t > c_t$ and $d_t = d_{t+1}$ and $c = c_{t+1} + 1$ and ($y'_s =$ blank and $s_{t+1} = s_t + 1$ or $s_{t+1} = s_t$).

IV. Evaluation

This section reports a comparative experiment conducted for evaluating the performance of the proposed ADT method and the effectiveness of the efficient training.

A. Experimental Conditions

The 100 songs of the RWC Music Database: Popular Music [31] were randomly split into 60 and 40 songs for training and test data, respectively. The stereo signals of each song sampled at 44.1 kHz were analyzed by short-time Fourier transform (STFT) with a window size of 1024 pts ($F = 513$) and a hop size of 441 pts (10 [ms]). The left and right channels were concatenated to form the input data $\mathbf{X}$. The tempo was assumed to be between 50 and 250 [bpm], i.e., the duration of a tatum was between $D_{\min} = 6$ to $D_{\max} = 30$.

We trained a convolutional recurrent neural network (CRNN) with the basic CTC or the proposed CTC2. It has 11 convolutional layers with a kernel size of 3×3, a padding size of 1×1, and a stride of 1 yielding a (512×4)-dimensional feature map, on which a dropout of 30% was applied before feeding it into a linear layer, followed by three bi-directional long short-term memory (BLSTM) layers with 200-dimensional hidden states. We used AdamW [32] with a learning rate $\gamma = 0.001$, a weight decay parameters $\lambda = 10^{-4}$, $\beta_1 = 0.9$, $\beta_2 = 0.98$, and $\varepsilon = 10^{-9}$ for optimization. When the Viterbi algorithm or the Gibbs sampling (Section III-E) was used with the CTC, the basic training (Section III-C) was performed for initialization. For the CTC2, the basic training could not be tested due to the huge computational cost.

For comparison, we implemented to test a recent audio-to-MIDI transcription method based on multi-task learning that jointly detects drum onset times and beat times at the frame level [2]. This method was trained using music signals with frame-level onset annotations, whereas our method was trained end-to-end using music signals with *non-aligned* scores. For MIDI-to-score transcription, the detected drum onset times were quantized on the tatum grid made of the beat times estimated by an HMM-based beat tracker [3] as post-processing.

Our end-to-end transcription method with the CTC or CTC2 and the cascading method mentioned above [2], [3] were evaluated in terms of the tatum- and frame-level accuracies. For tatum-level evaluation of estimated *scores*, we used the tatum error rate (TER), an edit distance similar in spirit to the word error rate (WER) in speech recognition, as follows:

$$\mathrm{TER} = \frac{\#\mathrm{insersion} + \#\mathrm{deletion} + \#\mathrm{substitution}}{N}, \tag{26}$$

where $\#$insersion, $\#$deletion, $\#$substitution represent the numbers of insertion, deletion, and substitution errors, respectively, and N represents the total number of tatums in the ground-truth data. For frame-level evaluation of estimated *onset times*, we set the error tolerance to 70 [ms]. Note that the initial frames of the active states (tatums) determined by the estimated alignment path $\boldsymbol{\pi}$ were regarded as drum onset times for convenience in our end-to-end method. Since those onset times tend to have a constant offset from the ground-truth onset times, the best offset was found for each song such that the performance was maximized for fair comparison.

TABLE I
THE TATUM- AND FRAME-LEVEL EVALUATION RESULTS.

Method	Tatum error rate ↓			F-measure ↑		
	Basic	Viterbi	Gibbs	Basic	Viterbi	Gibbs
CTC [14]	34.4	35.1	35.6	79.4	78.6	76.1
CTC2 (ours)	-	**33.8**	**33.8**	-	79.2	79.0
Vogl et al. [2]		34.5			**83.1**	

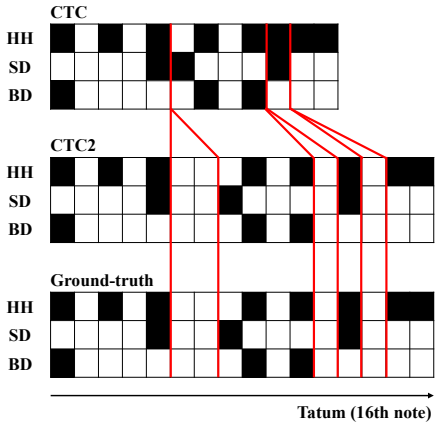

Fig. 5. Examples of tatum-level drum score fragments estimated by the CRNN trained with the basic CTC or the proposed CTC2. Whereas the CTC often failed to detect inactive states, the CTC2 correctly detected them.

B. Experimental Results

As shown in Table I, the CTC2 steadily outperformed the CTC in both metrics. Although the quantitative difference was small, the CTC2 yielded considerably better drum scores in the naturalness of rhythm despite the fact that the metrical structure was not considered. As shown in Fig. 5, the CTC2 successfully estimated a correct number of inactive tatums,

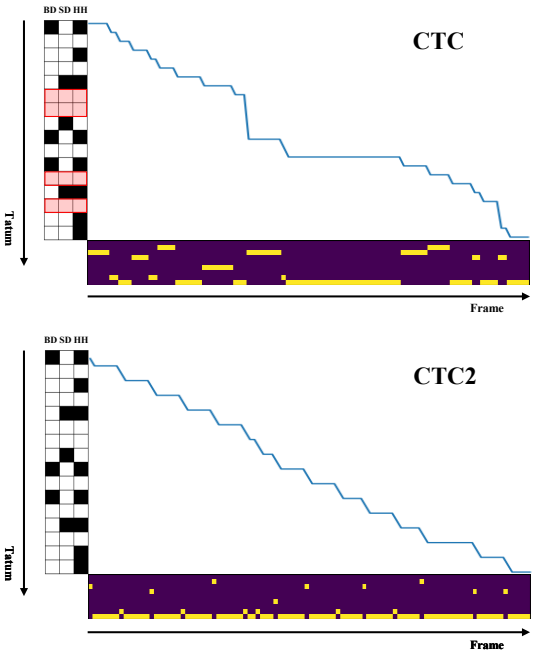

Fig. 6. Examples of alignment paths estimated by the CRNN trained with the basic CTC or the proposed CTC2 in the training phase. Whereas the CTC yielded a path with sudden tempo changes, the CTC2 yielded a diagonal path with an almost constant tempo.

which were missed by the CTC, between active tatums thanks to the constant tempo constraint. As shown in Fig. 6, in the training phase based on the forced alignment between the estimated and ground-truth state sequences, the CTC2 yielded an alignment path with an almost constant grade (tempo) around the diagonal line on the lattice. The CTC, in contrast, allowed an alignment path to discontinuously and frequently change the grade in a musically-unnatural manner. The Viterbi algorithm and the Gibbs sampling worked comparably in performance (Table I) and were several orders of magnitude faster than the basic backpropagation on the entire computation graph at a minimum sacrifice of performance (Fig. 7).

Compared with the conventional cascading method [2], our end-to-end method with the CTC2 performed slightly better by 0.7 pts in the TER, but worse by 6.0 pts in the F-measure. Note that the frame-level metric is advantageous for audio-to-MIDI transcription methods that use precise frame-level onset annotations for training. Considering that our method can be trained end-to-end with non-aligned audio-score pairs and does not aim at frame-level onset detection (the alignment path does not necessarily indicate precise onset times), this result is still considered to be promising.

V. Conclusion

In this paper, we proposed an end-to-end ADT method based on the CTC with constant tempo constraint, CTC2 in short, that estimates a sequence of tatum-level drum score fragments from a music signal. Although the loss function can be computed efficiently as in the vanilla CTC, the backpropagation

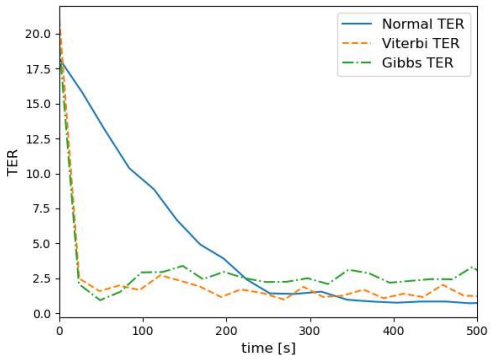

Fig. 7. Comparion of convergence speeds achieved by the basic training, the Viterbi algorithm, and the Gibbs learning.

over the huge computation graph made by the forward algorithm is computationally prohibitive. To solve this problem, we proposed a novel efficient training method that performs the backpropagation through only an alignment path found by the Viterbi algorithm or stochastically drawn with Gibbs sampling. Since non-aligned drum scores are easier to collect than labor-intensive frame-level onset annotations, the proposed end-to-end ADT method has a large potential for performance improvement based on large-scale training.

The CTC2 is a general technique for explicit duration modeling in a monotonic sequence-to-sequence mapping task not limited to ADT. It can be customized by formulating a duration model according to the task. In handwritten character recognition [33], for example, the CTC with constant "size" constraint would be useful for encouraging each character to have almost the same size. The Viterbi algorithm and the Gibbs sampling can also be applied to a wide range of tasks in various fields for accelerated CTC-based training. We are currently investigating how to effectively use the Viterbi algorithm in the inference phase as well as the training phase.

Acknowledgment

This study was partially supported by JSPS KAKENHI Nos. 21K12187, 21K02846, 22H03661, 20H00602, and 21H03572, JST PRESTO No. JPMJPR20CB, and JST FOREST No. JPMJPR226X.

References

[1] R. Stables, J. Hockman, and C. Southall. Automatic drum transcription using bi-directional recurrent neural networks. In *International Society for Music Information Retrieval Conference (ISMIR)*, 2016.

[2] R. Vogl, M. Dorfer, G. Widmer, and P. Knees. Drum transcription via joint beat and drum modeling using convolutional recurrent neural networks. In *International Society for Music Information Retrieval Conference (ISMIR)*, pp. 150–157, 2017.

[3] 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)*, pp. 72–78, 2015.

[4] S. Böck and M. E. P. Daveis. Deconstruct, analysis, reconstruct: How to improve tempo, beat, and downbeat estimation. In *International Society for Music Information Retrieval Conference (ISMIR)*, pp. 574–582, 2020.

[5] R. Ishizuka, R. Nishikimi, E. Nakamura, and K. Yoshii. Tatum-level drum transcription based on a convolutional recurrent neural network with language model-based regularized training. In *Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, 2020.

[6] 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.

[7] 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 *International Society for Music Information Retrieval Conference (ISMIR)*, pp. 50–57, 2018.

[8] E. Nakamura, E. Benetos, K. Yoshii, and S. Dixon. Towards complete polyphonic music transcription: Integrating multi-pitch detection and rhythm quantization. In *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 101–105. IEEE, 2018.

[9] M. A. Román, A. Pertusa, and J. Calvo-Zaragoza. An end-to-end framework for audio-to-score music transcription on monophonic excerpts. In *International Society for Music Information Retrieval Conference (ISMIR)*, pp. 34–41, 2018.

[10] R. C. G. Carvalho and P. Smaragdis. Towards end-to-end polyphonic music transcription: Transforming music audio directly to a score. In *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pp. 151–155. IEEE, 2017.

[11] R. Nishikimi, E. Nakamura, M. Goto, and K. Yoshii. End-to-end melody note transcription based on a beat-synchronous attention mechanism. In *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pp. 26–30. IEEE, 2019.

[12] R. Nishikimi, E. Nakamura, S. Fukayama, M. Goto, and K. Yoshii. Automatic singing transcription based on encoder-decoder recurrent neural networks with a weakly-supervised attention mechanism. In *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 161–165. IEEE, 2019.

[13] L. Liu, V. Morfi, and E. Benetos. Joint multi-pitch detection and score transcription for polyphonic piano music. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 281–285. IEEE, 2021.

[14] A. Graves, S. Fernandez, F. Gomez, and J. Schmidhuber. Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. In *International conference on Machine Learning (ICML)*, pp. 369–376, 2006.

[15] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. In *International Conference on Learning Representations (ICLR)*, pp. 1–15, 2015.

[16] M. Russell and R. Moore. Explicit modelling of state occupancy in hidden markov models for automatic speech recognition. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Vol. 10, pp. 5–8. IEEE, 1985.

[17] E. Nakamura, K. Yoshii, and S. Sagayama. Rhythm transcription of polyphonic piano music based on merged-output HMM for multiple voices. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, Vol. 25, No. 4, pp. 794–806, 2017.

[18] C. Jacques and A. Roebel. Automatic drum transcription with convolutional neural networks. In *International Conference on Digital Audio Effects (DAFx)*, 2018.

[19] C. Jacques and A. Roebel. Data augmentation for drum transcription with convolutional neural networks. In *European Signal Processing Conference (EUSIPCO)*, pp. 1–5. IEEE, 2019.

[20] S. Ueda, K. Shibada, Y. Wada, R. Nishikimi, E. Nakamura, and K. Yoshii. Drum transcription using convolutional non-negative matrix factorization based on deep drum score prior distribution (in japanese). *IPSJ SIG technical reports (EC)*, Vol. 2019, No. 26, pp. 1–6, 2019.

[21] Y. Wang, J. Salamon, M. Cartwright, Nicholas J. Bryan, and J. P. Bello. Few-shot drum transcription in polyphonic music. In *International Society for Music Information Retrieval Conference (ISMIR)*, pp. 117–124, 2020.

[22] R. Ishizuka, R. Nishikimi, and K. Yoshii. Global structure-aware drum transcription based on self-attention mechanisms. *Signals*, Vol. 2, No. 3, pp. 508–526, 2021.

[23] D. Kamakura, T. Oyama, and K. Yoshii. Drum transcription and beat structure estimation based on multi-task learning (in japanese). *The 84th National Convention of IPSJ*, Vol. 2022, No. 1, pp. 517–518, 2022.

[24] C. Lea, R. Vidal, A. Reiter, and G. D. Hager. Temporal convolutional networks: A unified approach to action segmentation. In *European Conference on Computer Vision (ECCV) Workshop*, pp. 47–54, 2016.

[25] M. E. P. Daveis and S. Böck. Temporal convolutional networks for musical audio beat tracking. In *European Signal Processing Conference (EUSIPCO)*, pp. 1–5. IEEE, 2019.

[26] M. Cartwright and J. P. Bello. Increasing drum transcription vocabulary using data synthesis. In *International Conference on Digital Audio Effects (DAFx)*, pp. 72–79, 2018.

[27] K. Choi and K. Cho. Deep unsupervised drum transcription. In *International Society for Music Information Retrieval Conference (ISMIR)*, pp. 183–191, 2019.

[28] R. G. C. Carvalho and P. Smaragdis. Towards end-to-end polyphonic music transcription: Transforming music audio directly to a score. In *Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pp. 151–155. IEEE, 2017.

[29] M. A. Román, A. Pertusa, and J. Calvo-Zaragoza. An end-to-end framework for audio-to-score music transcription on monophonic excerpts. In *International Society for Music Information Retrieval Conference (ISMIR)*, pp. 34–41, 2018.

[30] J. Van Gael, Y. Saatci, Yee Whye Y. W. Teh, and Z. Ghahramani. Beam sampling for the infinite hidden Markov model. In *International conference on Machine Learning (ICML)*, pp. 1088–1095, 2008.

[31] 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)*, pp. 287–288, 2002.

[32] I. Loshchilov and F. Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, pp. 1–8, 2017.

[33] M. Ibrayim, W. Simayi, and A. Hamdulla. Unconstrained online handwritten uyghur word recognition based on recurrent neural networks and connectionist temporal classification. *International Journal of Biometrics*, Vol. 13, No. 1, pp. 51–63, 2021.