Local Periodicity-Based Beat Tracking for Expressive Classical Piano Music

Ching-Yu Chiu, Meinard Müller, Matthew E. P. Davies, Alvin Wen-Yu Su, *Member, IEEE*, and Yi-Hsuan Yang, *Senior Member, IEEE*

***Abstract*—To model the periodicity of beats, state-of-the-art beat tracking systems use "post-processing trackers" (PPTs) that rely on several empirically determined global assumptions for tempo transition, which work well for music with a steady tempo. For expressive classical music, however, these assumptions can be too rigid. With two large datasets of Western classical piano music, namely the Aligned Scores and Performances (ASAP) dataset and a dataset of Chopin's Mazurkas (Maz-5), we report on experiments showing the failure of existing PPTs to cope with local tempo changes, thus calling for new methods. In this paper, we propose a new local periodicity-based PPT, called predominant local pulse-based dynamic programming (PLPDP) tracking, that allows for more flexible tempo transitions. Specifically, the new PPT incorporates a method called "predominant local pulses" (PLP) in combination with a dynamic programming (DP) component to jointly consider the locally detected periodicity and beat activation strength at each time instant. Accordingly, PLPDP accounts for the local periodicity, rather than relying on a global tempo assumption. Compared to existing PPTs, PLPDP particularly enhances the recall values at the cost of a lower precision, resulting in an overall improvement of F1-score for beat tracking in ASAP (from 0.473 to 0.493) and Maz-5 (from 0.595 to 0.838).**

***Index Terms*—Beat tracking, expressive music, post-processing tracker.**

Manuscript received 23 April 2022; revised 16 November 2022, 26 February 2023, and 3 July 2023; accepted 4 July 2023. Date of publication 21 July 2023; date of current version 28 July 2023. The work of Meinard Müller was supported by the International Audio Laboratories Erlangen – a joint institution of the Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU) and Fraunhofer Institute for Integrated Circuits IIS. The work of Matthew E. P. Davies was supported in part by the FCT–Foundation for Science and Technology, I.P., through the Project MERGE through the National Funds (PIDDAC) through the Portuguese State Budget under Grant PTDC/CCI-COM/3171/2021, and in part by the European Social Fund through the Regional Operational Program Centro 2020 Project CISUC under Grant UID/CEC/00326/2020. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Isabel Barbancho. *(Corresponding author: Ching-Yu Chiu.)*

Ching-Yu Chiu is with the Graduate Program of Multimedia Systems and Intelligent Computing, National Cheng Kung University and Academia Sinica, Tainan 701401, Taiwan (e-mail: sunnycyc@citi.sinica.edu.tw).

Meinard Müller is with the International Audio Laboratories Erlangen, 91058 Erlangen, Germany (e-mail: meinard.mueller@audiolabs-erlangen.de).

Matthew E. P. Davies is with the Department of Informatics Engineering, Centre for Informatics and Systems, University of Coimbra, 3004-531 Coimbra, Portugal (e-mail: mepdavies@dei.uc.pt).

Alvin Wen-Yu Su is with the Department of Computer Science and Information Engineering, National Cheng Kung University, Tainan 701401, Taiwan (e-mail: alvinsu@mail.ncku.edu.tw).

Yi-Hsuan Yang is with Yating Music Team, Taiwan AI Labs, Taipei 103622, Taiwan. He is now with Research Center for IT Innovation, Academia Sinica, Taipei 11529, Taiwan (e-mail: yang@citi.sinica.edu.tw).

Digital Object Identifier 10.1109/TASLP.2023.3297956

I. Introduction

BEATS, which are generally referred to as a sequence of perceived pulses at the temporal level that a listener would tap to, are fundamental to the understanding of music [1], [2]. The ability to perceive beats not only allows us to follow the music, but also serves as the basis for decomposing, reconstructing, or interacting with music. Computationally, a variety of downstream applications are related to, and could be enhanced by beat tracking [3], [4].

Existing beat tracking systems are mainly composed of two parts. First, a *novelty detection* module generates a so-called "novelty function" (sometimes also called "activation function"), which is a continuous-valued curve that captures the energy or spectral changes over time so as to reveal beat candidates. Second, a *post-processing tracker* (PPT) gives the final binary decision regarding beat occurrence [5], [6], [7]. While traditional model-based systems usually derive the novelty function based on onset detection methods [8], [9], [10], [11], [12], more recent deep learning (DL)-based systems use a feature-learning network to compute directly from audio signals an activation function, indicating the likelihood of observing a beat at each time instant. Due to the fact that existing novelty detection methods do not handle well the periodic nature of beat times [13], existing beat tracking systems generally rely on periodicity-aware PPTs to determine the beat positions. Motivated by the design of [6], [14], most existing PPTs are state-space models (see Section II-B for more details), such as dynamic Bayesian network (DBN), hidden Markov model (HMM) [6], [13], [14], conditional random field (CRF) [15], or particle filtering [16].

DL-based beat tracking systems have achieved great success for music with steady tempo, in particular for pop, rock, and dance music [6], [7], [17]. However, probably due to the scarcity of publicly available data, the performance of state-of-the-art DL-based beat trackers for expressive classical music has seldomly been discussed, or is far from satisfactory if reported. Specifically, the performance of state-of-the-art DL-based systems for beat or downbeat tracking tends to be 20–30% worse for classical music than for other music genres [18], [19], [20]. In this article, we are interested in finding out the underlying reasons for the poor performance while improving beat tracking for expressive classical music.

The challenges of beat tracking for expressive classical music are closely related to the properties of the novelty function [21], [22]. For example, the novelty function may get aperiodic due

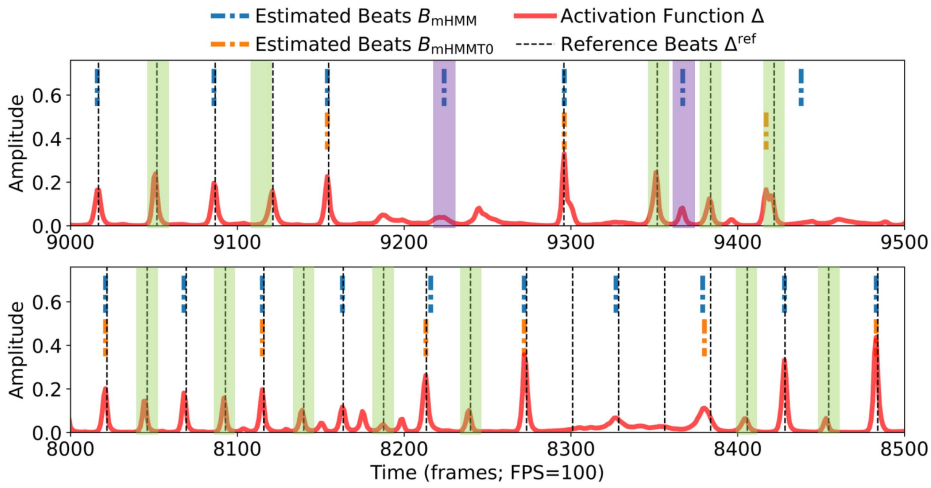

Fig. 1. Beat tracking result of a state-of-the-art DL-based system from madmom for two recordings of classical music, one from Maz-5 [21] **(top)** and the other from ASAP [24] (**bottom**). The activation function (using a frame rate of FPS = 100) is shown in red. Green shades highlight the false-negative estimations with activation peaks. Purple shades highlight the false-positive estimations caused by non-beat activation peaks. Best viewed in color.

to substantial tempo, rhythmic, or note density variations. Its intensity may get weaker due to blurred note onsets and soft note transitions for non-percussive instruments such as violins or singing. It is also assumed that one may improve the accuracy of beat tracking for classical music by using a larger classical music training set [18], [23]. As one contribution of this article, we present experiments on two large datasets of expressive classical piano music, the Chopin Mazurkas (Maz-5) [21] and the Aligned Scores and Performances (ASAP) dataset [24], and show that there are fundamental issues related to the PPT that needs to be addressed first.

Fig. 1 illustrates the result of beat tracking for two recordings of classical music, one from Maz-5 and the other from ASAP, using the system from the madmom library [6], [17]. The system uses an ensemble of recurrent neural networks (RNN) for estimating the beat activation functions, and an HMM-based PPT [14] for ensuring the periodicity of the detected beats. As mentioned, existing PPTs for beat tracking closely follow the HMM-based PPT in madmom library [6], [14]. As such, we adopt this PPT as one of our main baseline models, using the default (and widely-used) parameter setting of madmom. We refer to this PPT as "mHMM" hereafter. Besides, in our study we also consider a more flexible version of mHMM where we tune its parameter setting to allow for a higher tempo transition likelihood, and refer to it as "mHMMT0" (see Sections IV-B and IV-C for details). We can see from the reference beats (i.e., the ground truth annotations) that the two recordings feature different degrees of tempo variation. We also see that the system has many false positives and false negatives, caused not only by its imperfect beat activation function (e.g., activation peaks at non-beat positions), but also by the HMM-based PPT. For both recordings, mHMM assumes that there is a relatively stable (and slower) tempo and chooses to ignore local activation peaks corresponding to true beat positions. Similar detection errors can be observed from the result of mHMMT0. Moreover, our analysis (see Section IV-D) shows that, even with a *perfect* oracle activation function created synthetically with the reference beats, the average F1 score obtained by the mHMM-based estimations across the 301 recordings of Maz-5 remains lower than 0.80. This suggests that the PPT makes some tempo assumptions that do not work for expressive music.

In cognitive neuroscience, it has been found that the human brain generally focuses more on local events (e.g., musical onsets within a small time window) and continuously tries to predict incoming information [25]. Given a few musical onset events, we may start having expectations for the incoming events [26], [27], [28]. These expectations are to be adjusted, strengthened, or abandoned based on the consistency between the expectations and the incoming events. These "temporal expectations" may be part of the reasons why humans can, to a certain degree, adaptively deal with tempo variations, syncopation, and rest notes on beat positions in music.

In light of the above observations, we develop a new PPT method that mimics the temporal expectations computed from the beat activation function. Specifically, we propose to use a feature called "predominant local pulses" (PLP) [8], [29], [30] (see Section III for details) to estimate information regarding local periodicity (analogous to temporal expectations) from the beat activation function. The PLP curve is converted into two curves, one of which contains information of local inter-pulse-intervals (IPIs) and the other containing the confidence of locally detected periodicity. Then, we propose a new dynamic programming (DP)-based PPT that takes the two curves as time-varying tempo conditions to track the beats. With quantitative experiments, using both real and synthetic activation functions on Maz-5 and ASAP (Section V), we demonstrate the advantages of the new PPT method, named "PLPDP," over representative existing PPTs for beat tracking of expressive music with continuously varying tempo. We also identify some limitations of PLPDP that need to be further addressed in future work (Section VI).[1]

[1]For reproducibility, we provide open-source code for PLPDP at: https://github.com/SunnyCYC/plpdp4beat/. We also have a project web page that provides examples of the beat tracking results: https://sunnycyc.github.io/plpdp4beat-demo/.

II. Related Works

A. Beat and Tempo for Expressive Classical Music

Research on beat tracking for classical music with large tempo variations dates back two decades [2], [31]. However, it was not until the work by Grosche et al. [8], [21] that larger evaluations were carried out. In their first publication [21], they discussed five musical properties that cause problems for beat tracking, and subsequently conducted systematic experiments to analyze and make the limitations of state-of-the-art beat trackers explicit. As beat tracking systems at that time relied more on the quality of the underlying novelty function, the influence of different musical properties on the novelty detection was illustrated. However, the influence of musical properties on the performance of PPT was not explicitly investigated. In the publication [8], the idea of predominant local pulse (PLP) was introduced and used for modeling the local periodicity of model-based onset novelty function. Specifically, PLP extracts and enhances the local periodicity of the input novelty function via analyzing the onset peaks within small windows, which motivates the core idea of this article (see Section III). The PLP-enhanced novelty function can be combined with a PPT based on dynamic programming (DP), e.g., as introduced by Ellis [1]. However, assuming an overall constant tempo (see Section II-B), the DP-based PPT cannot handle strong tempo variations.

More recently, motivated by the work of Böck et al. [6], [17], a variety of DL-based feature learning networks have been proposed [7], [32]. Moreover, as DL models typically require large training data, only a few studies tackle beat tracking-related tasks for classical music where beat annotations are hardly available. For example, Schreiber et al. [23] pioneered the use of DL-based methods for modeling local tempo of Maz-5. Specifically, they aggregated several beats into a higher-level local tempo representation, and used that local tempo representation as the target of their DL-based model. In other words, their work aims at estimating the local tempo, but not for predicting the individual beats.

B. PPTs With Global Assumptions for Tempo

The dynamic programming-based beat tracker (DP) as introduced by Ellis [1] is a widely-used PPT that we consider as a baseline in our evaluation. The DP method assumes that the musical piece is performed with a roughly constant tempo and that the activation is high at beat positions. DP introduces a score that jointly considers how well an input novelty function fits the two assumptions and finds globally the best beat sequence that maximizes the score via a dynamic programming algorithm. Based on the constant tempo assumption, a global tempo value is used to balance out the consistency between target and estimated inter-beat-interval (IBI) (see Section III-D for details). The global tempo value can be derived either from the mean IBI of the reference beats [8], or via auotocorrelation-based tempo estimation methods [1], [33].

While the constant tempo assumption grants DP a simple formulation and realization, it also limits the PPT's flexibility. Aiming at jointly modeling tempi and beats, Krebs et al. [14] extended the "bar pointer model" [34], [35], and proposed a refined state-space discretization and tempo transition model. Their main contributions are the design of a state-space discretization model that ensures sufficient tempo resolution for each hidden state and time resolution consistency between hidden states of different tempi, and a new transition model based on a first-order Markov assumption to improve the stability of tempo trajectories. In particular, they increased the tempo stability by proposing a transition model that only allows tempo transitions at beat positions and empirically adopted an exponential distribution function as the tempo transition likelihood function (see Section IV-B3 for details).

Due to their success in both reducing computational cost and outperforming the original model, existing mainstream PPTs [6], [13], [15], [16], [36], [37] are mainly motivated by [14]. These PPTs may differ in their optimization mechanisms [36] or the use of extra information (e.g., beat phase [13] or time signature [6]), but they generally adopt similar empirically determined tempo transition likelihood functions based on a first-order Markov assumption.

We note that in related studies on rhythm transcription, researchers also apply HMM-based methods to take an input signal and estimate the metrical positions of all musical notes. In particular, similar first-order Markov assumptions and tempo transition probability distributions are adopted [38], [39], [40]. Despite that local tempo and local tempo changes are parameterized in these models, parameters are determined globally (e.g., based on a dataset) without the knowledge of "local periodicity" (which is explicitly extracted based on small local windows) proposed in this work. Focusing on beat tracking for expressive classical piano music, we limit our discussion of HMMs to those proposed for beat tracking and not for the considered HMMs for rhythm transcription.

III. PLPDP-Based PPT

In this section, we provide the details of the proposed PLPDP-based PPT. We first introduce the PLP concept [8], [29], [30] and then elaborate on the similarity between PLP and human temporal expectations. Next, we describe a method to reduce the artifacts of PLP curves at regions with tempo variations. Subsequently, we demonstrate how to derive the tempo-related information reflecting local temporal expectation and confidence from a PLP function. Finally, we present the algorithm that connects DP with the two tempo-related conditions to realize the PLPDP-based PPT.

A. PLP as Local Temporal Expectations

As mentioned in Section I, we found that the PLP, a method originally proposed to model and enhance the periodicity of musical onset novelty functions [8], shows behaviors that are interestingly akin to human temporal expectations [41]. Fig. 2 demonstrates the computation and the "temporal expectations" of PLP given a pre-computed novelty function. The main idea of PLP is to generate periodic pulses that align with a given novelty curve based on a local periodicity assumption. This is

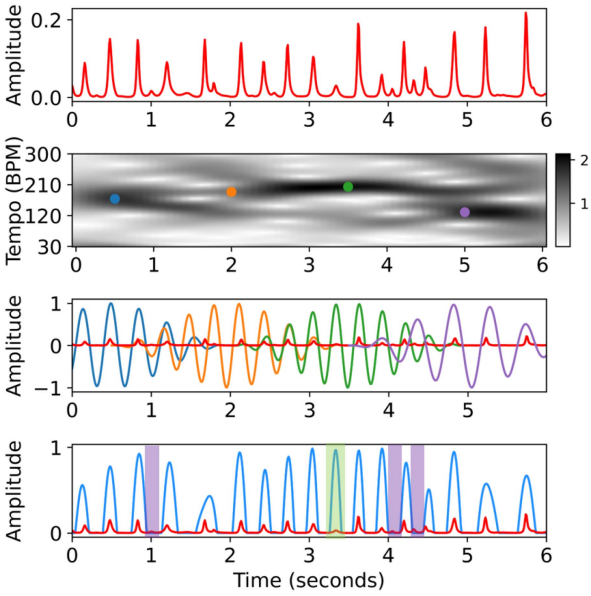

Fig. 2. Illustration of the workflow of the PLP calculation. **(a)** The novelty function (red curve; also plotted in (c) and (d) with smaller amplitude for alignment comparison) computed from a recording. **(b)** Fourier tempogram with four colored dots indicating the corresponding time positions of the optimal sinusoidal kernels in (c). **(c)** Optimal sinusoidal kernels at four different time positions. **(d)** PLP (blue curve) derived via overlap-adding and half-wave rectification. Purple shades highlight the novelty peaks suppressed by PLP. Green shade highlights the novelty peak enhanced by PLP.

achieved by locally comparing the novelty curve with windowed sinusoidal kernels, and accumulating over time all the optimal sinusoidal kernels that best capture the local peak structure of the novelty function. Specifically, the computation of PLP begins with computing the "Fourier tempogram" [42] (cf. Fig. 2(b)) via a short-time Fourier transform (STFT), for a certain pre-determined tempo range (e.g., $\theta \in [30:300]$ beats-per-minute; BPM). Then, given pre-determined values of the STFT parameters, kernel size κ and hop size h, the predominant tempo and phase information of the optimal sinusoidal kernel of each time instant can be derived from the tempogram and the underlying complex-valued Fourier coefficients [43]. For example, Fig. 2(c) shows the optimal sinusoidal kernels at four time positions with $\kappa = 3$ (seconds). Finally, via overlap-adding and half-wave rectification (keeping only positive parts of the curve), the PLP curve can be derived. Note that the peaks of PLP somehow indicate PLP's "expectation" of the existence of novelty peaks (e.g., see Fig. 2(d) how the PLP peaks align with the peaks of the novelty function). Even if some novelty peaks are low, as long as the locally detected periodicity has high confidence, PLP still generates strong peaks (see the peak underlined with green shade in Fig. 2(d)). This is similar to how humans can still tap on weak or even rest notes based on temporal expectations. Furthermore, novelty peaks that do not match the detected local periodicity are suppressed (see the regions indicated by purple shades). Moreover, as exemplified in Fig. 2(d), PLP has lower peaks (being less confident) when the neighboring musical events are not consistent in periodicity. Inspired by the human ability to adaptively adjust expectations and confidence concerning

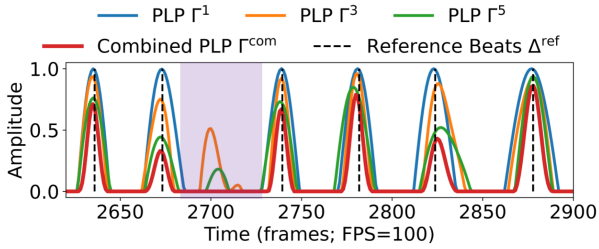

Fig. 3. PLP functions with different kernel sizes $\kappa = 1$ (blue), $\kappa = 3$ (orange), and $\kappa = 5$ (green), and the combined PLP function (red), computed using the oracle novelty function created from the reference beats Δ^{ref} (dashed vertical lines). Purple shade highlights the region with a larger tempo/IBI variation where extra peaks of individual PLP functions are suppressed by the combined PLP function. Best viewed in color.

current and future events, we incorporate the idea of PLP in our beat trackers to mimic this ability.

B. PLP Curves and Combination

The "local sensitivity" of a PLP function is largely determined by the choice of the kernel size κ. Let $\Gamma^{\kappa} : [1:N] \to [0,1]$ denote the PLP function for kernel size κ (given in seconds), where $[1:N] := \{1, 2, \ldots, N\}$, $N \in \mathbb{N}$, represents the sampled time axis with respect to a fixed sampling rate. In our experiments, we use a rate of 100 frames per second. As an example, Fig. 3 shows Γ^{κ} for three different kernel sizes $\kappa \in \{1, 3, 5\}$, using the oracle novelty function $\Delta^{\text{ref}} : [1:N] \to [0,1]$ created from its reference beats as input.[2] It can be seen that at regions with relatively stable tempo, PLP curves with different kernel sizes generally have similar "temporal expectations" for peak positions, though with different confidences (i.e., peak heights). However, for regions with a larger tempo variation, the three PLP functions show incosistent behaviors (e.g., see the region shaded in purple in Fig. 3). As the PLP curves based on different kernel sizes typically show artifacts at different time positions, we find combining these PLP curves by element-wise multiplication a simple yet effective way of reducing the artifacts. We therefore define the combined PLP function Γ^{com} by

$$\Gamma^{\text{com}}(n) := \Gamma^{1}(n) \cdot \Gamma^{3}(n) \cdot \Gamma^{5}(n) \tag{1}$$

for $n \in [1:N]$.

C. PLP as Tempo-Related Condition

PLP functions yield *peak positions* that are aligned with peaks in the input novelty function while the *peak heights* can be regarded as a measure of confidence. To obtain local tempo-related information, we propose the following procedure to convert a PLP function into a piecewise constant function λ expressing confidence and a piecewise constant function $\hat{\delta}$ encoding inter-beat-intervals (IBIs). As exemplified in Fig. 4, we

[2]PLP can take as input either a real or synthetic novelty function (cf. Sections IV-C and IV-D). We use a synthetic one here (created from reference beats; see Section IV-D for details) to show the sensitivity of PLP to tempo variations.

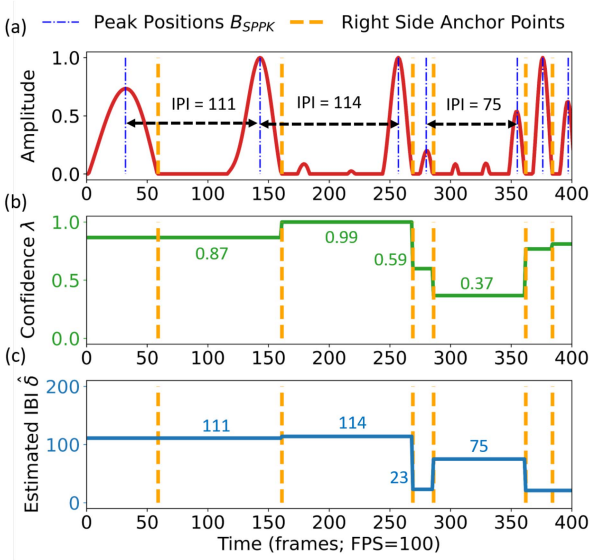

Fig. 4. Conversion of a PLP function into piecewise constant functions of confidence λ and estimated IBI $\hat{\delta}$. **(a)** Division of a PLP function into segments using right side anchor points of detected peaks. **(b)** Confidence function derived from peak heights. **(c)** Estimated IBI function $\hat{\delta}$ derived from inter-peak-intervals (IPIs).

first apply a simple *peak picking* function (SPPK)[3] to the PLP to obtain a list of peak positions $B_{\text{SPPK}} = (b_1, b_2, \ldots, b_K)$ (marked by blue lines in Fig. 4(a)). We then divide the PLP into a number of segments at time instances corresponding to the right side anchor points of these peaks (marked by vertical orange lines).[4] For each such segment, we compute the inter-peak-interval (IPI) and set $\hat{\delta}(n)$ to this value for all frames n within the segment, see Fig. 4(c). Similarly, we define $\lambda(n)$ to be the average height of the segment's two PLP peaks. We show below how the resulting two functions, confidence and estimated IBI, can be employed by a PPT for tracking the beats in expressive music.

D. Combination of PLP and DP

The DP-based beat tracker introduced in [1] aims at finding the optimal beat sequence B^* that maximizes a score function C balancing out novelty intensity and tempo consistency. Let $B = (b_1, b_2, \ldots, b_K)$ be a sequence of estimated beat positions in chronological order. The score C function is defined by:

$$C(B) := \sum_{k=1}^{K} \Delta(b_k) + \lambda_0 \sum_{k=2}^{K} P_{\hat{\delta}_0}(b_k - b_{k-1}), \qquad (2)$$

where Δ denotes the novelty function, $\lambda_0 \in \mathbb{R}_{\geq 0}$ denotes a factor to balance the relative importance of the novelty function and

[3]In our implementation, we use `scipy.signal.find_peaks` [44] with parameters `height` $= 0.1$, `distance` $= 7$, and `prominence` $= 0.1$. The distance value of seven frames is set to correspond to the tolerance window size (i.e., 70 ms) for beat tracking evaluation, and the other two values are set to 0.1 to ensure basic height and prominence of peaks.

[4]While there might be other more complicated methods, we chose this simple heuristic that divides the PLP curve at the right side anchor points (i.e., positions where the curve reaches a low value after a peak).

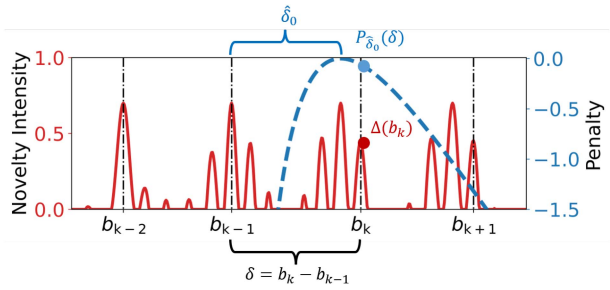

Fig. 5. Illustration of the score function $C(B)$ defined in (2), which jointly shows the novelty function intensity $\Delta(b_k)$ and the tempo consistency penalty function $P_{\hat{\delta}_0}(\delta)$.

the tempo consistency condition. Furthermore, $P_{\hat{\delta}_0} : \mathbb{N} \to \mathbb{R}$ denotes a penalty function for tempo consistency with respect to a preassigned IBI $\hat{\delta}_0 \in \mathbb{N}$ defined by

$$P_{\hat{\delta}_0}(\delta) := -\left(\log_2\left(\frac{\delta}{\hat{\delta}_0}\right)\right)^2 \qquad (3)$$

for $\delta = b_k - b_{k-1}$. Note that $P_{\hat{\delta}_0}(\delta)$ is large for $\delta \approx \hat{\delta}_0$ and decreases for smaller or larger δ values. See Fig. 5 for an illustration of $P_{\hat{\delta}_0}$ and the definition of the score function C.

The original DP [1], [45], [29, Section 6.3.2] takes a fixed IBI $\hat{\delta}_0$ and a fixed factor λ_0. In this article, we propose a new DP-based PPT, named PLPDP, that takes the novelty function δ and the time-varying values $\hat{\delta}(n)$ and $\lambda(n)$ introduced in Section III-C as input. Compared to the original DP, we replace $\hat{\delta}_0$ by $\hat{\delta}(n)$, and replace λ_0 by $\lambda(n)$. In the DP *forward* procedure, we compute at time frame n an accumulated score $\mathbf{D}(n)$ which depends on the accumulated scores of the previous frames $m \in [1 : n-1]$, the corresponding tempo consistency penalties, and current novelty intensity:

$$\mathbf{D}(n) = \Delta(n) + \max\left\{0, \max_{m \in [1:n-1]}\left\{\mathbf{D}(m) + \lambda(n) P_{\hat{\delta}(n)}(n-m)\right\}\right\}.$$

At the same time, we save in $\mathbf{P}(n)$ the predecessor time position of the maximum. $\mathbf{D}(n)$ and $\mathbf{P}(n)$ are then used in the *backward* procedure to derive the optimal beat sequence B^*. Algorithm 1 shows the pseudo-code of PLPDP.[5]

We consider as the default case the combined PLP function Γ^{com} as input for PLPDP. To empirically justify the use of multiple kernels, we also consider in our experiments an ablation study, where we use only the PLP function Γ^3 with a single kernel, dubbed "PLPDP-Γ^3."

IV. Experiment Setup

In the following, we report on experiments to evaluate previous state-of-the-art and our proposed PPTs, see Section V. In these experiments, we consider two datasets (Maz-5, ASAP), see Section IV-A. Furthermore, we consider activation functions

[5]This algorithm is modified based on the DP algorithm in [29] pp.343, Table 6.1. and [46].

Algorithm 1 PLPDP Beat Tracking.

INPUT
– novelty (activation) function $\Delta : [1:N] \to [0,1]$
– confidence $\lambda : [1:N] \to [0,1]$ and
estimated IBI $\hat{\delta} : [1:N] \to \mathbb{R}_{\geq 0}$ derived from PLP
OUTPUT
– optimal beat sequence $B^* = (b_1, b_2, \ldots, b_K)$
– accumulated score $\mathbf{D}(n)$
– predecessor information $\mathbf{P}(n)$
PROCEDURE
Forward :
Initialize $\mathbf{D}(0) = 0$ and $\mathbf{P}(0) = 0$.
Then compute in a loop for $n = 1, \ldots, N$:
$\mathbf{D}(n) = \Delta(n) +$
$\max\{0, \max_{m\in[1:n-1]}\{\mathbf{D}(m) + \lambda(n) P_{\hat{\delta}(n)}(n-m)\}\}$
If $\mathbf{D}(n) = \Delta(n)$ then set $\mathbf{P}(n) = 0$,
otherwise set
$\mathbf{P}(n) =$
$\arg\max_{m\in[1:n-1]}\{\mathbf{D}(m) + \lambda(n) P_{\hat{\delta}(n)}(n-m)\}$
Backward :
Set $k = 1$ and $a_k = \arg\max_{n\in[0:N]} \mathbf{D}(n)$
Then repeat the following steps until $\mathbf{P}(a_k) = 0$:
Increase k by one.
Set $a_k = \mathbf{P}(a_{k-1})$.
If $a_k = 0$, then set $K = 0$ and **return** $B^* = \emptyset$.
Otherwise let $K = k$ and
return $B^* = (a_K, a_{K-1}, \ldots, a_1)$.

TABLE I
STATISTICS OF DATASETS USED IN THE EXPERIMENTS

Dataset	# tracks	Total duration	% of stable tempi
Maz-5	301	12h 27m	13.1%
ASAP	502	41h 45m	24.6%

computed from audio recordings (real use case) and obtained from GT annotations (synthetic scenario).

A. Statistics of Datasets

Table I lists the two datasets employed in this study. Maz-5 [21] is a private collection of music composed of 301 audio recordings corresponding to five of the 49 different Chopin Mazurkas. These recordings were collected as part of the Mazurka project [47] and manually annotated (beat positions) by Sapp [48]. ASAP [24] is a public dataset newly released in 2020, consisting of 502 performances of Western classical piano music from 15 composers.[6]

Table I also shows the *tempo stability rate* for the datasets, calculated according to the approach of Schreiber [23]. Specifically, we first convert all IBIs of a dataset into tempo values, and divide these tempo values by the average tempo of the corresponding track to derive normalized tempi. With the commonly adopted $\pm 4\%$ tolerance interval for quantifying whether the tempi of a recording are stable or not [50], we calculate the percentage of recordings in a dataset whose normalized tempi falls between the 0.96–1.04 interval. Table I shows that the tempo stability rate of Maz-5 and ASAP is 13.1% and 24.6%, respectively, suggesting that a large portion of the recordings in both datasets do not have a steady tempo.[7]

B. Baseline/Proposed PPTs

We consider in our experiments four baselines PPTs(SPPK, DP, mHMM, and mHMMT0) as well as our procedures, PLPDP, and PLPDP-Γ^3. In the following, we describe these PPTs in more detail.

1) Simple Peak-Picking (SPPK): This procedure applies the `find_peaks` function from the `scipy` library [44] to detect peak positions in the activation function. These peaks are taken as beats without imposing assumptions such as those regarding the beat periodicity consistency, and without any corrections of the input (e.g., novelty function enhancement).[3]

2) DP+GT: As mentioned in Section III-D, DP from [1] requires a pre-assigned IBI $\hat{\delta}_0$ as global tempo information, which could be estimated by a global tempo detection approach [52]. However, as the focus here is to investigate the intrinsic limitations of DP, we use the ground-truth (GT) global tempo derived from the mean IBI of the reference beat positions. Note that using GT tempo values gives DP advantages over the other uninformed PPTs thus avoiding error propagation from imperfect global tempo estimation. Moreover, we will show in our experiments that, even with access to some GT information, this DP+GT baseline does not perform well for Maz-5 and ASAP, due to its intrinsic limitations. In our experiments, we use GT-informed DP following the settings of Grosche et al. [8].[8]

3) mHMM and mHMMT0: As another baseline approach, we employ the classic HMM-based PPT proposed by Krebs et al. [14], [17].[9] The main components of this approach are a state-space discretization and tempo transition model. Given an input novelty function $\Delta : [1:N] \to [0,1]$ as the observation sequence, the tempi to be considered are represented by a set

$$\mathcal{A} := \{\alpha_1, \alpha_2, \ldots, \alpha_I\} \tag{4}$$

of size $I \in \mathbb{N}$ consisting of distinct elements α_i for $i \in [1:I]$. The elements α_i are referred to as hidden tempo states determined by the discretization methods and a given tempo range. The tempo transition can be realized by a system that can be described at any time instance $n \in [1:N]$ as being in one of the tempo states $\dot{\Psi}_n \in \mathcal{A}$.

Kreb et al. [14] proposed a tempo transition model that mostly stays in the same tempo and only allows tempo changes at beat positions. For a time instance n that corresponds to a

[6]During the execution of this project, a dataset called ACPAS [49] that combines ASAP and another 59 real recordings of classical piano performances was released. We have the analysis and evaluation for those relatively small number of recordings not included in this work.

[7]In comparison, as reported [23], the percentage of recordings with stable tempi reaches 90.9% for the *Ballroom* dataset [51], a collection of dance music widely used in research on beat and downbeat tracking.

[8]Source codes of DP can be found in [45].

[9]We adopted the official released `DBNBeatTrackingProcessor` of `madmom`, which does not consider the time signature of input tracks.

beat position, they empirically adopt the following exponential distribution function as the tempo change likelihood function:

$$f(\dot{\Psi}_n, \dot{\Psi}_{n-1}) = \exp\left(-\lambda_{\text{trans}} \cdot \left|\frac{\dot{\Psi}_n}{\dot{\Psi}_{n-1}} - 1\right|\right), \quad (5)$$

where the tempo transition lambda $\lambda_{\text{trans}} \in \mathbb{R}_{\geq 0}$ determines the steepness of the distribution.[10] Intuitively speaking, using a large parameter λ_{trans} makes the model rigid, only allowing small tempo changes from beat to beat. Conversely, a small parameter λ_{trans} (close to zero) makes a transition to all possible tempi almost equally likely.

By default, the tempo transition lambda λ_{trans} in (5) for mHMM is empirically chosen as $\lambda_{\text{trans}} = 100$ based on empirical observations from mainstream pop, rock, or dance music. We denote the default one as mHMM and refer to [14] for further details. To allow for a much larger tempo variation, as encountered in classical music, we implement a variant of the mHMM with $\lambda_{\text{trans}} = 0$ and denote it as mHMMT0. To further reveal the capability and limitation of the HMM-based PPTs, experiments of grid search for λ_{trans} are reported and discussed in Section V-D.

4) PLPDP and PLPDP-Γ^3: There are several options for implementing the PLPDP. For example, the hop size h, the kernel size κ, or design choices for extracting tempo-related conditions from PLP, could all be optimized in some way (e.g., grid search). However, as our focus is on investigating the general behaviors of these local temporal-based methods rather than optimizing these methods for a specific type of dataset, we omit such optimization/fine-tuning processes. We empirically set $\kappa = 1, 3, 5$ seconds for the combined PLP (Γ^{com}) as input of PLPDP, and $\kappa = 3$ seconds for the PLP (Γ^3) as input of PLPDP-Γ^3 used in our ablation study. We set the tempo range to $[60 : 300]$ (given in BPM) for $\kappa = 1$ and to $[30 : 300]$ (given in BPM) for $\kappa = 3, 5$ to ensure that each PLP kernel can at least accommodate one completed sinusoidal wave for each given tempo.

C. Real Use Case

In the real use case, we take original audio recordings as the input and derive for each recording the beat activation and downbeat activation (indicating probability of beat and downbeat at each frame) via `RNNDownBeatProcessor` of `madmom` [6], [17], which is a DL-based approach. We then take the maximum of the two activation functions at each frame to derive a joint beat activation function as input of the various PPTs.

D. Synthetic Scenario

Rather than computing activation functions from audio recording, we also consider a synthetic scenario where using idealized activation functions derived from ground-truth beat annotations. To this end, we transform the annotation into a pulse train of equal pulse magnitudes using a frame rate of 100 FPS. This way, the input activation functions are "perfect" as they are all of the maximum strength (set to $1 - \epsilon$) at beat positions and with minimum values (set to ϵ) at non-beat positions.[11] We expect the synthetic experiments based on synthetic activation functions to reveal the sensitivity of the PPTs to tempo stability without the influence caused by errors in the estimated activation functions.

[10] Note that the tempo discretization ($\dot{\Psi}$) of mHMM state-space is nonlinear and different from θ (in Section III-A). Readers may refer to [14] for details. In this work, we set the tempo range of both mHMMs and PLPDP as 30–300 BPM. I.e., $\theta \in [30 : 300]$, and $(\texttt{min\_bpm}, \texttt{max\_bpm}) = (30, 300)$ for `DBNBeatTrackingProcessor` of `madmom`.

TABLE II
BEAT TRACKING RESULT ON THE MAZ-5 DATASET

PPT	Real activation			Synthetic activation		
	F1	Recall	Precision	F1	Recall	Precision
SPPK	**0.822**	0.754	**0.918**	**1.000**	**1.000**	**1.000**
DP+GT	0.488	0.501	0.475	0.799	0.808	0.791
mHMM	0.499	0.393	0.753	0.794	0.872	0.732
mHMMT0	0.595	0.450	**0.903**	**0.994**	0.994	**0.995**
PLPDP-Γ^3	0.791	**0.936**	0.696	0.862	**1.000**	0.766
PLPDP	**0.838**	**0.917**	0.777	0.982	0.996	0.968

The two best scores per metric are highlighted in bold.

V. EXPERIMENT RESULTS

In this section, we report on our experiment results for the real use case and synthetic scenario. We use a tolerance window of $\pm$ 70 ms to calculate the recall (R), precision (P), and F-measure (F1) as the performance metrics.

A. Quantitative Result on Maz-5

We start our discussion with the Maz-5 dataset. Table II shows the beat tracking results for various settings.

For the real activation case, we first look at the results for SPPK to get some insights regarding the properties of the Maz-5 dataset and corresponding activation functions. As SPPK picks all activation peaks as beat positions, we can infer from the high precision value ($P = 0.918$) that most activation peaks correspond to beat positions and infer from the recall value ($R = 0.754$) that there are some missing peaks in Maz-5. Next, note that the baseline PPTs (e.g., DP and mHMM) can hardly achieve an F1 score higher than 0.6, which is in contrast to their superior performance reported in references [6], [7], [8] for music with steady tempo. From the lower recall values of DP and HMM-based methods (DP: $R = 0.501$, mHMM: $R = 0.393$, mHMMT0: $R = 0.450$) compared to SPPK ($R = 0.754$), we can further see that their poor performance is mainly due to ignorance of activation peaks. Besides, from the precision values of DP ($P = 0.475$) and mHMM ($P = 0.753$) which are lower than SPPK ($P = 0.918$), we can see that DP and mHMM insert beat estimations at positions without activation peaks based on their strict tempo assumptions.

On the other hand, PLPDP behaves differently than other PPTs. From the remarkably high recall (PLPDP-Γ^3: $R = 0.936$, PLPDP: $R = 0.917$) compared to SPPK ($R = 0.754$), we see the effectiveness of "local temporal expectations" to compensate

[11] A small value $\epsilon = 10^{-6}$ is needed to prevent error warnings of the `madmom` API [17] for our implementation of mHMM.

TABLE III
BEAT TRACKING RESULT ON THE ASAP DATASET

PPT	Real activation			Synthetic activation		
	F1	Recall	Precision	F1	Recall	Precision
SPPK	0.380	0.419	**0.607**	**1.000**	**0.999**	**1.000**
DP+GT	0.450	0.458	0.443	0.903	0.913	0.894
mHMM	0.473	0.540	0.500	0.911	0.947	0.886
mHMMT0	0.374	0.324	**0.556**	0.982	0.986	**0.981**
PLPDP-Γ^3	**0.488**	**0.732**	0.404	0.829	**0.997**	0.750
PLPDP	**0.493**	**0.707**	0.418	**0.982**	0.995	0.971

The two best scores per metric are highlighted in bold.

for the missing activation peaks at beat positions. On the contrary, from the lower precision values (PLPDP-Γ^3: $P = 0.696$, PLPDP: $P = 0.777$) compared to SPPK ($P = 0.918$), we see that PLPDP also make false-positive estimations based on its local temporal expectations. Overall, the above behavioral differences between PLPDP and existing PPTs lead to a substantial performance gap of F1 score (PLPDP: $F1 = 0.838$ vs. mHMMT0: $F1 = 0.595$ and mHMM: $F1 = 0.499$), indicating the effectiveness of "local temporal periodicity" for Maz-5. Additionally, the high F1 score of SPPK ($F1 = 0.822$) implies that for an expressive music recording with high-quality activation functions (e.g., with fewer number of non-beat activation peaks), SPPK may achieve a high F1-score.

The results of the synthetic case provide further insights into the above observations. Note that taking the perfect synthetic activation functions as input, one may expect all PPTs to achieve an F1 score of 1.0. However, except for SPPK (the only PPT without any tempo-related assumptions or restrictions), none of the other PPTs could achieve so. From the imperfect recall and precision, we can see that the strong global tempo assumptions of DP and mHMM not only lead to discarding activation peaks, but also introduce false-positive beat predictions in regions without any activation peaks. Observing these inherent limitations, the poor performance of DP and HMM in the real activation experiments may be less surprising.

It can also be seen from the higher recall and precision of PLPDP that the proposed method can adapt better to the local tempo variations of Maz-5 when the input activation is perfect. Accordingly, PLPDP is more flexible than DP and mHMM.[12] Moreover, the higher precision of PLPDP compared to PLPDP-Γ^3 suggests the effectiveness of a combined PLP function rather than a single-kernel PLP function.

B. Quantitative Result on ASAP

Table III shows the beat tracking results for the ASAP dataset. From the substantial performance change of SPPK, we can conclude that there are significant differences between the properties of ASAP and Maz-5. Both recall (ASAP: $R = 0.419$, Maz-5: $R = 0.754$) and precision (ASAP: $P = 0.607$, Maz-5: $P = 0.918$) values drop dramatically. This indicates that, for ASAP, the `madmom` network fails to generate activation peaks at some beat positions while generating a large number of spurious activation peaks. Such differences may be caused by the properties of datasets (e.g., ASAP may contain more non-beat note events) or insufficient training of the DL-based `madmom` network (not adapted to ASAP). These observations further explain the results that none of the PPTs can achieve an F1 score higher than 0.50, which is different from the result for Maz-5. We recall that the tempo stability of ASAP is higher than Maz-5, as shown in Table I. Therefore, the results for ASAP indicate that the poor beat tracking performance may also be caused by the properties of activation functions.

Despite the above mentioned differences between the two datasets, similar behavioral patterns of PLPDP can be observed. From the higher recall of PLPDP compared to the other PPTs, we can see again the effectiveness of the "local temporal expectations" to compensate for the missing peaks at beat positions of activation functions. However, from the lower precision of PLPDP in ASAP compared to Maz-5, one may deduce that with an increasing number of activation peaks at non-beat positions, the performance of PLPDP and PLPDP-Γ^3 drop substantially.

Similarly, the results of the synthetic case for ASAP support the above observations. The obvious fact that SPPK has perfect scores for all three metrics[13] again reflects how PPTs are limited by their intrinsic tempo assumptions. Comparing the F1 scores between Maz-5 and ASAP in the synthetic case, we also see that DP and mHMM are sensitive to low tempo stability and perform much better as the tempo gets more stable in ASAP. In contrast, PLPDP is less sensitive to tempo changes and performs similarly for both datasets.

C. Qualitative Results

Fig. 6 shows the beat tracking result of PLPDP for two real activation functions (continuing the examples from Fig. 1). The reference beats and activation functions (Fig. 6, top row) illustrate the observations mentioned before. For example, the Maz-5 recording has more tempo variations and reveals fewer activation peaks at non-beat positions. On the other hand, we see several weak or missing activation peaks at beat positions of the ASAP example. The PLP functions for $\kappa = 1, 3, 5$ (Fig. 6, middle row) further reveal the different temporal expectations at regions with tempo changes. The combined PLP function and estimated beats of PLPDP (Fig. 6, bottom row) reveal the advantages and limitations of the local temporal expectations. Specifically, PLPDP nicely adapts for both examples to the local tempo variations based on its local temporal expectations. However, these expectations may also cause false-positive errors if there are activation peaks at non-beat positions that match the locally

[12] One may argue that mHMMT0 outperforms PLPDP in this synthetic case of Maz-5. We would like to note that the main purpose of synthetic experiments is to investigate the limitations of the assumptions of each PPT. With the most flexible setting of lambda, mHMMT0 is indeed more flexible than PLPDP (though with difference <1.2%). However, this flexible setting also remarkably limits the performance of mHMMT0 in real (imperfect) activation cases. Besides, we note that when the optimal activation function is achievable (e.g., the DL-based networks really learn the comprehensive idea of beats as humans), the best PPT is always SPPK (i.e., without any assumption).

[13] Its recall rate is not 1.0, mostly due to annotation errors (i.e., reference beats that are too close to each other and excluded by SPPK's `distance` = 7 setting.) caused by the semi-automatic annotating process of ASAP [24].

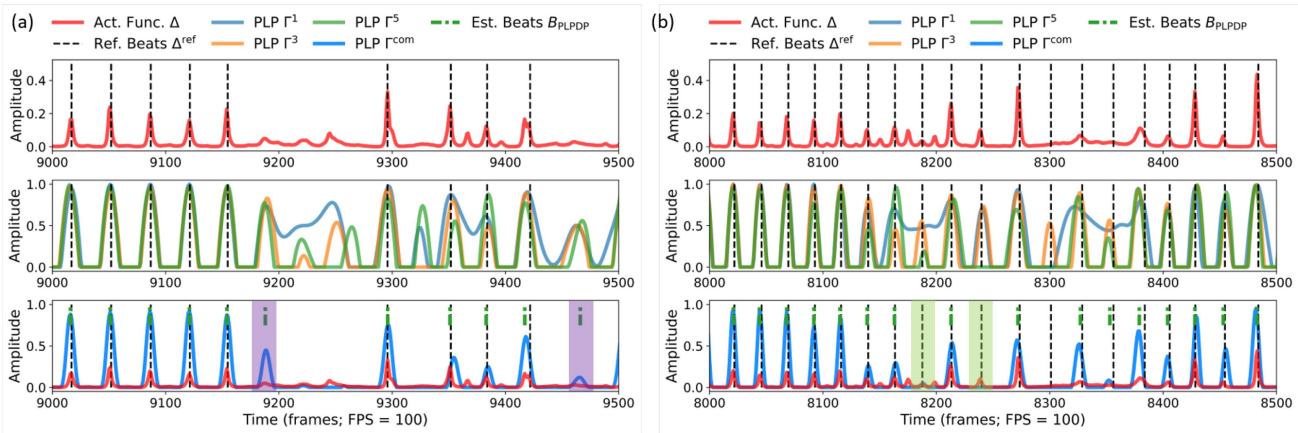

Fig. 6. Beat tracking result of the proposed PLPDP-based PPT for the same two recordings used in Fig. 1. **(a)** Maz-5 example. **(b)** ASAP example. **Top:** `madmom` activation function and reference beats. **Middle:** PLP functions with $\kappa = 1, 3, 5$. **Bottom:** Combined PLP function and estimated beats of PLPDP. Purple-shaded regions highlight the false-positive estimations. Green-shaded regions highlight false-negative estimations. Best viewed in color.

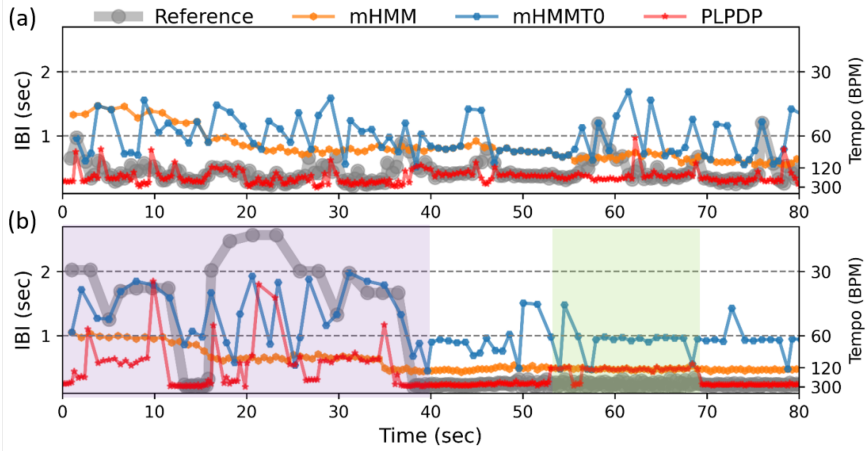

Fig. 7. IBI progression of reference beats (grey) and estimated beats (blue: mHMM, orange: mHMMT0, red: PLPDP) for the recordings of the excerpts used in Fig. 1. **(a)** Maz-5 recording. **(b)** ASAP recording. Purple-shaded region highlights a region with slow unstable tempi. Green-shaded region highlights a region with fast stable tempi which PPTs fail to follow.

detected periodicity, as highlighted by the the purple-shaded regions. From false-negative errors (green-shaded regions), we see that PLPDP may also ignore activation peaks at beat positions based on its local temporal expectations. However, as long as most of the activation peaks are stronger at beat positions, PLPDP introduces much less such false-negative errors than mHMM.

Fig. 7 demonstrates the longer-term behavior of the PPTs for the recordings of the two examples via plotting the inter-beat-interval (IBI) progression. Specifically, for each sequence of beat positions (e.g., reference beats or PPT-based estimated beats), we include the beat positions b_i (in horizontal axis given in seconds) and its corresponding IBI (i.e., the value $b_{i+1} - b_i$ plotted on vertical axis) to see the reference/estimated IBI progression within the recordings. We can see from the reference (grey) curve that while the Maz-5 example (Fig. 7(a)) reveals continuous tempo changes at faster tempi (i.e., 100–300 BPM, corresponding to IBIs between 0.2–0.6 seconds.), the ASAP example (Fig. 7(b)) has both an slow unstable region (shaded in purple) and a region with faster stable tempi (shaded in green). For both pieces, any global assumption without explicitly considering the local musical contents is not likely to work well. Explicitly, as the tempo transition function of mHMMs are set in a global manner, both mHMMs are not able to align with the reference IBI progression like PLPDP can do.

For more qualitative results, we refer the readers to the project web page.[1]

D. Grid Search of mHMM Tempo Transition λ_{trans}

The above experiments have already demonstrated the conceptual difference between HMM-based PPTs and the proposed PLPDP approach. We now present an additional grid-search experiment with regard to the mHMM tempo transition parameter λ_{trans} to provide some additional insights. Fig. 8 shows the results for real the activation case. While the mHMM approache with lambda λ_{trans} varying from 1 to 25 performs better than mHMMT0 for Maz-5, the best mHMM still performs worse

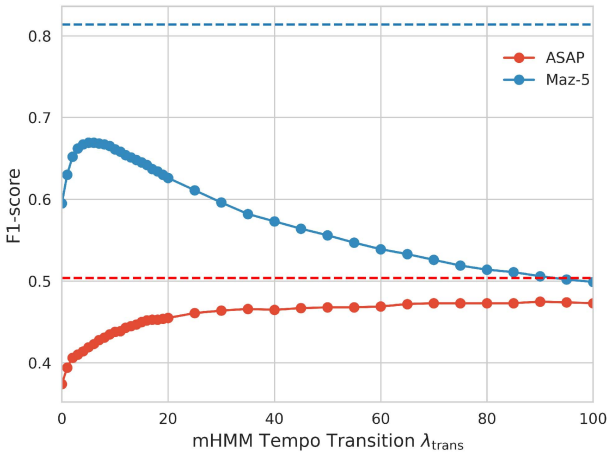

Fig. 8. Grid search of mHMM tempo transition λ_{trans} from 0–100 for real activation experiments. A step size of one is adopted for $0 \leq \lambda_{\text{trans}} \leq 20$, and five otherwise. Results of PLPDP are indicated as horizontal dashed lines for comparison.

than PLPDP (shown as horizontal dashed lines) for both datasets. Furthermore, though both datasets consist of expressive classical music, the best performing λ_{trans} are different (i.e., Maz-5: $\lambda_{\text{trans}} = 5$, ASAP: $\lambda_{\text{trans}} = 90$). One may therefore deduce that adjusting the parameter λ_{trans} for each recording and even to local sections within a recording may be essential to improve the performance of the mHMM approach.

VI. Conclusion and Future Work

In this article, we have made contributions towards improving and better understanding beat tracking for expressive classical music. First, we introduced a new local temporal expectation-based post processing tracking (PPT) method. Second, we conducted experiments to investigate the performance upper bounds of the considered PPTs. Third, we presented a comprehensive evaluation and analysis of beat tracking approaches for expressive classical music. The proposed PLPDP method provides a way to incorporate local tempo-related information into a beat tracking system. Considering local periodicity consistency, our method differs from existing PPTs that rely on globally determined tempo transition assumptions. Moreover, our synthetic experiments demonstrate new ways to investigate and explore the strength and limitations of PPTs. Overall, we hope this work provides a new direction towards improving beat tracking for expressive classical music.

From the real activation experiments on ASAP, we see that the PPTs still have a large margin for improvements. Among the factors that influence the performance of beat tracking, the missing peaks of activation at beat positions could be potentially reduced by adding more data of classical music for training the feature-learning networks. In particular, this may lead to substantial improvements of activation functions that can better account for the various properties of note onsets as occurring in classical music.

Adding training data alone, however, might not help the DL-based networks to produce less (false-positive) non-beat activation peaks, which also constitute a large part of the beat tracking errors in particular for ASAP. Instead of relying completely on onset-related information, we conjecture that approaches that consider hierarchical cues, e.g., frequency domain information as pitch, melody, or longer-term structure-related information, might prove useful. Besides, these hierarchically arranged musical and acoustic cues may also be helpful for future models to adaptively adjust PLP kernel sizes.

From our empirical observations of the behavior of all the considered PPTs, we find that these procedures often switch between different metric-levels (e.g., half, third, double or triple tempo of reference beats) while tracking the beats of a recording. Such a "metric-level switching" behavior, however, cannot be reflected by the current evaluation metrics. We have recently proposed an analysis method [53] for gaining a better understanding of such issues. More results and discussions can be found in our GitHub repository.

Lastly, as existing datasets of multi-instrument classical music (e.g., RWC-Classical [54]) are relatively small, we consider only Western classical piano music in our experiments. Compared to piano music, there might be more onsets at non-beat positions in multi-instrument classical music, and the activation intensity at beat positions may be weaker due to soft onsets. To assess the performance of PLPDP for expressive classical music in general, future work is needed to consider datasets beyond piano music.

References

[1] D. P. Ellis, "Beat tracking by dynamic programming," *J. New Music Res.*, vol. 36, no. 1, pp. 51–60, 2007.
[2] S. Dixon and E. Cambouropoulos, "Beat tracking with musical knowledge," in *Proc. Eur. Conf. Artif. Intell.*, 2000, pp. 626–630.
[3] E. Benetos, S. Dixon, Z. Duan, and S. Ewert, "Automatic music transcription: An overview," *IEEE Signal Process. Mag.*, vol. 36, no. 1, pp. 20–30, Jan. 2019.
[4] Y.-S. Huang and Y.-H. Yang, "Pop Music Transformer: Beat-based modeling and generation of expressive pop piano compositions," in *Proc. ACM Int. Conf. Multimedia*, 2020, pp. 1180–1188.
[5] M. Fuentes, B. McFee, H. C. Crayencour, S. Essid, and J. P. Bello, "Analysis of common design choices in deep learning systems for downbeat tracking," in *Proc. Int. Soc. Music Inf. Retrieval Conf.*, 2018, pp. 106–112.
[6] S. Böck, F. Krebs, and G. Widmer, "Joint beat and downbeat tracking with recurrent neural networks," in *Proc. Int. Soc. Music Inf. Retrieval Conf.*, 2016, pp. 255–261.
[7] S. Böck and M. E. P. Davies, "Deconstruct, analyse, reconstruct: How to improve tempo, beat, and downbeat estimation," in *Proc. Int. Soc. Music Inf. Retrieval Conf.*, 2020, pp. 574–582.
[8] P. Grosche and M. Müller, "Extracting predominant local pulse information from music recordings," *IEEE Trans. Audio, Speech Lang. Process.*, vol. 19, no. 6, pp. 1688–1701, Aug. 2011.
[9] J. P. Bello, L. Daudet, S. Abdallah, C. Duxbury, M. Davies, and M. B. Sandler, "A tutorial on onset detection in music signals," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 13, no. 5, pp. 1035–1047, Sep. 2005.
[10] R. Zhou, M. Mattavelli, and G. Zoia, "Music onset detection based on resonator time frequency image," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 16, no. 8, pp. 1685–1695, Nov. 2008.
[11] A. Klapuri, "Sound onset detection by applying psychoacoustic knowledge," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 6, pp. 3089–3092, 1999.
[12] A. P. Klapuri, A. J. Eronen, and J. T. Astola, "Analysis of the meter of acoustic musical signals," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 14, no. 1, pp. 342–355, Jan. 2006.
[13] T. Oyama, R. Ishizuka, and K. Yoshii, "Phase-aware joint beat and downbeat estimation based on periodicity of metrical structure," in *Proc. Int. Soc. Music Inf. Retrieval Conf.*, 2021, pp. 493–499.

[14] F. Krebs, S. Böck, and G. Widmer, "An efficient state-space model for joint tempo and meter tracking," in *Proc. Int. Soc. Music Inf. Retrieval Conf.*, 2015, pp. 72–78.
[15] M. Fuentes, B. McFee, H. C. Crayencour, S. Essid, and J. P. Bello, "A music structure informed downbeat tracking system using skip-chain conditional random fields and deep learning," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2019, pp. 481–485.
[16] M. Heydari, F. Cwitkowitz, and Z. Duan, "BeatNet: CRNN and particle filtering for online joint beat downbeat and meter tracking," in *Proc. Int. Soc. Music Inf. Retrieval Conf.*, 2021, pp. 270–277.
[17] S. Böck, F. Korzeniowski, J. Schlüter, F. Krebs, and G. Widmer, "Madmom: A new Python audio and music signal processing library," in *Proc. ACM Int. Conf. Multimedia*, 2016, pp. 1174–1178.
[18] C.-Y. Chiu, A. W.-Y. Su, and Y.-H. Yang, "Drum-aware ensemble architecture for improved joint musical beat and downbeat tracking," *IEEE Signal Process. Lett.*, vol. 28, pp. 1100–1104, 2021.
[19] C.-Y. Chiu, J. Ching, W.-Y. Hsiao, Y.-H. Chen, A. W.-Y. Su, and Y.-H. Yang, "Source separation-based data augmentation for improved joint beat and downbeat tracking," in *Proc. 29th Eur. Signal Process. Conf.*, 2021, pp. 391–395.
[20] S. Durand and S. Essid, "Downbeat detection with conditional random fields and deep learned features," in *Proc. Int. Soc. Music Inf. Retrieval Conf.*, 2016, pp. 386–392.
[21] P. Grosche, M. Müller, and C. S. Sapp, "What makes beat tracking difficult? A case study on Chopin Mazurkas," in *Proc. Int. Soc. Music Inf. Retrieval Conf.*, 2010, pp. 649–654.
[22] A. Holzapfel, M. E. P. Davies, J. R. Zapata, J. L. Oliveira, and F. Gouyon, "Selective sampling for beat tracking evaluation," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 20, no. 9, pp. 2539–2548, Nov. 2012.
[23] H. Schreiber, F. Zalkow, and M. Müller, "Modeling and estimating local tempo: A. case study on Chopin's Mazurkas," in *Proc. Int. Soc. Music Inf. Retrieval Conf.*, 2020, pp. 773–779.
[24] F. Foscarin, A. McLeod, P. Rigaux, F. Jacquemard, and M. Sakai, "ASAP: A dataset of aligned scores and performances for piano transcription," in *Proc. Int. Soc. Music Inf. Retrieval Conf.*, 2020, pp. 534–541.
[25] A. Clark, "Whatever next? predictive brains, situated agents, and the future of cognitive science," *Behav. Brain Sci.*, vol. 36, no. 3, pp. 181–204, 2013.
[26] F. L. Bouwer, H. Honing, and H. A. Slagter, "Beat-based and memory-based temporal expectations in rhythm: Similar perceptual effects, different underlying mechanisms," *J. Cogn. Neurosci.*, vol. 32, no. 7, pp. 1221–1241, 2020.
[27] J. Obleser, M. Henry, and P. Lakatos, "What do we talk about when we talk about rhythm ?," *PLOS Biol.*, vol. 15, 2017, Art. no. e2002794.
[28] A. Nobre and F. Ede, "Anticipated moments: Temporal structure in attention," *Nature Rev. Neurosci.*, vol. 19, pp. 34–48, 2017.
[29] M. Müller, *Fundamentals of Music Processing–Using Python and Jupyter Notebooks*, 2nd ed. Berlin, Germany: Springer, 2021.
[30] P. Meier, G. Krump, and M. Müller, "A real-time beat tracking system based on predominant local pulse information," in *Proc. Demos Late Breaking News Int. Soc. Music Inf. Retrieval Conf.*, 2021. [Online]. Available: https://www.bibsonomy.org/bibtex/26817f45ddc4a3603b86844eaf107bf92/baywiss1
[31] S. Dixon, "An empirical comparison of tempo trackers," in *Proc. Braz. Symp. Comput. Music*, 2001, pp. 832–840.
[32] E. P. Matthew Davies and S. Böck, "Temporal convolutional networks for musical audio beat tracking," in *Proc. 27th Eur. Signal Process. Conf.*, 2019, pp. 1–15.
[33] M. E. P. Davies and M. Plumbley, "Context-dependent beat tracking of musical audio," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 15, no. 3, pp. 1009–1020, Mar. 2007.
[34] N. Whiteley, A. Cemgil, and S. Godsill, "Bayesian modelling of temporal structure in musical audio," in *Proc. Int. Soc. Music Inf. Retrieval Conf.*, 2006, pp. 29–34.
[35] N. Whiteley, A. T. Cemgil, and S. Godsill, "Sequential inference of rhythmic structure in musical audio," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2007, pp. 1321–1324.
[36] M. Heydari and Z. Duan, "Don't look back: An online beat tracking method using RNN and enhanced particle filtering," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2021, pp. 236–240.
[37] K. Yamamoto, "Human-in-the-loop adaptation for interactive musical beat tracking," in *Proc. Int. Soc. Music Inf. Retrieval Conf.*, 2021, pp. 794–801.
[38] K. Shibata, E. Nakamura, and K. Yoshii, "Non-local musical statistics as guides for audio-to-score piano transcription," *Inf. Sci.*, vol. 566, pp. 262–280, 2021.
[39] E. Nakamura, E. Benetos, K. Yoshii, and S. Dixon, "Towards complete polyphonic music transcription: Integrating multi-pitch detection and rhythm quantization," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2018, pp. 101–105.
[40] E. Nakamura, N. Ono, S. Sagayama, and K. Watanabe, "A stochastic temporal model of polyphonic MIDI performance with ornaments," *J. New Music Res.*, vol. 44, pp. 287–304, 2014.
[41] Z. Xu et al., "Temporal expectation driven by rhythmic cues compared to that driven by symbolic cues provides a more precise attentional focus in time," *Attention Perception Psychophys.*, vol. 83, pp. 308–314, 2021.
[42] P. Grosche, M. Müller, and F. Kurth, "Cyclic tempogram–a mid-level tempo representation for musicsignals," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2010, pp. 5522–5525.
[43] M. Müller, "Predominant local pulse (PLP)," 2021. [Online]. Available: https://www.audiolabs-erlangen.de/resources/MIR/FMP/C6/C6S3_PredominantLocalPulse.html
[44] P. Virtanen et al., "SciPy 1.0: Fundamental algorithms for scientific computing in Python," *Nature Methods*, vol. 17, pp. 261–272, 2020.
[45] M. Müller and F. Zalkow, "libfmp: A Python package for fundamentals of music processing," *J. Open Source Softw.*, vol. 6, no. 63, 2021, Art. no. 3326.
[46] M. Müller, "Beat tracking by dynamic programming," 2021. [Online]. Available: https://www.audiolabs-erlangen.de/resources/MIR/FMP/C6/C6S3_BeatTracking.html
[47] The mazurka project. 2010. [Online]. Available: http://mazurka.org.uk
[48] C. Sapp, "Hybrid numeric/rank similarity metrics for musical performance analysis," in *Proc. Int. Soc. Music Inf. Retrieval Conf.*, 2008, pp. 501–506.
[49] L. Liu, V. Morfi, and E. Benetos, "ACPAS dataset: Aligned classical piano audio and score," in *Proc. Demos Late Breaking News Int. Soc. Music Inf. Retrieval Conf.*, 2021. [Online]. Available: https://zenodo.org/record/5569680
[50] F. Gouyon et al., "An experimental comparison of audio tempo induction algorithms," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 14, no. 5, pp. 1832–1844, Sep. 2006.
[51] F. Krebs, S. Böck, and G. Widmer, "Rhythmic pattern modeling for beat and downbeat tracking in musical audio," in *Proc. Int. Soc. Music Inf. Retrieval Conf.*, 2013, pp. 227–232.
[52] H. Schreiber, J. Urbano, and M. Müller, "Music tempo estimation: Are we done yet ?," *Trans. Int. Soc. Music Inf. Retrieval*, vol. 3, no. 1, pp. 111–125, 2020.
[53] C.-Y. Chiu, M. Müller, M. E. P. Davies, A. W.-Y. Su, and Y.-H. Yang, "An analysis method for metric-level switching in beat tracking," *IEEE Signal Process. Lett.*, vol. 29, pp. 2153–2157, 2022.
[54] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka, "RWC music database: Popular, classical, and jazz music databases," in *Proc. Int. Soc. Music Inf. Retrieval Conf.*, 2002, pp. 287–288.

Ching-Yu Chiu received the Ph.D degree from the Graduate Program of Multimedia Systems and Intelligent Computing, National Cheng Kung University and Academia Sinica, Tainan, Taiwan, in 2023. She is currently a Postdoctoral Researcher with International Audio Laboratories Erlangen, Erlangen, Germany, a joint institute of the Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU), Erlangen, and the Fraunhofer Institute for Integrated Circuits IIS, Erlangen. Her research interests include music information retrieval, signal processing, and machine learning.

Meinard Müller received the Diploma degree in mathematics and the Ph.D. degree in computer science from the University of Bonn, Bonn, Germany, in 1997 and 2001, respectively. After his postdoctoral studies during 2001–2003 in Japan and his habilitation during 2003–2007 in multimedia retrieval in Bonn, he was a Senior Researcher with Saarland University, Saarbrcken, Germany, and the Max-Planck Institut für Informatik during 2007–2012. Since 2012, he has Professor of semantic audio signal processing with the International Audio Laboratories Erlangen, Erlangen, Germany, a joint Institute of the Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU), Erlangen, and the Fraunhofer Institute for Integrated Circuits IIS, Erlangen. His recent research interests include music processing, music information retrieval, audio signal processing, and motion processing. He was a Member of the IEEE Audio and Acoustic Signal Processing Technical Committee during 2010–2015, a Member of the Senior Editorial Board of the *IEEE Signal Processing Magazine* during 2018–2022, and a Member of the Board of Directors, International Society for Music Information Retrieval during 2009–2021, being its president in 2020 and 2021. In 2020, he was elevated to IEEE Fellow for contributions to music signal processing.

Matthew E. P. Davies received the B.Eng. degree in computer systems with electronics from King's College London, London, U.K., in 2001, and the Ph.D. degree in electronic engineering from the Queen Mary University of London (QMUL), London, in 2007. From 2007 to 2011, he was a Postdoctoral Researcher with the Centre for Digital Music, QMUL. In 2013, he worked in the Media Interaction Group, National Institute of Advanced Industrial Science and Technology. From 2014 to 2019, he coordinated the Sound and Music Computing Group, INESC TEC. He is currently a Researcher with the Centre for Informatics and Systems of the University of Coimbra, Coimbra, Portugal. His main research interests include music information retrieval, evaluation methodology, and creative music systems.

Alvin Wen-Yu Su (Member, IEEE) received the B.S. degree in control engineering from National Chiao-Tung University, Hsinchu, Taiwan, in 1986, and the M.S. and Ph.D. degrees in electrical engineering from Polytechnic University, Brooklyn, NY, USA, in 1990 and 1993, respectively. From 1993 to 1994, he was with the Center for Computer Research in music and acoustics with Stanford University, Stanford, CA, USA. He is currently a Professor of the Department of Computer Science and Information Engineering, National Cheng-Kung University, Tainan, Taiwan. His research interests include the areas of physical modeling of acoustic musical instruments, data compression, audio/image/video signal processing, and VLSI.

Yi-Hsuan Yang (Senior Member, IEEE) received the Ph.D. degree in communication engineering from National Taiwan University, Taipei, Taiwan. Since 2023, he has been with the College of Electrical Engineering and Computer Science, National Taiwan University, where he is currently a Full Professor. Prior to that, he was the Chief Music Scientist with an Industrial Lab called Taiwan AI Labs from 2019 to 2023, and an Associate/Assistant Research Fellow of the Research Center for IT Innovation, Academia Sinica, from 2011 to 2023. His research interests include automatic music generation, music information retrieval, and machine learning. He was an Associate Editor for IEEE TRANSACTIONS ON AFFECTIVE COMPUTING and the IEEE TRANSACTIONS ON MULTIMEDIA both from 2016 to 2019.