dPLP: A DIFFERENTIABLE VERSION OF PREDOMINANT LOCAL PULSE ESTIMATION

Ching-Yu Chiu, Sebastian Strahl, and Meinard Müller
International Audio Laboratories Erlangen, Germany
{ching-yu.chiu, sebastian.strahl, meinard.mueller}@audiolabs-erlangen.de

ABSTRACT

Predominant Local Pulse (PLP) estimation is a key technique in rhythmic analysis of music recordings, designed to identify the most salient pulse in an audio signal while adapting to local tempo variations. Unlike global tempo estimation, which assumes a fixed tempo, PLP dynamically adjusts to changes in tempo and rhythm, making it particularly effective as a post-processing strategy to enhance the locally periodic structure of a given input novelty or activity function. Traditional PLP estimation relies on a max operation to select the most prominent periodicity, limiting its use in differentiable learning frameworks. In this paper, we introduce dPLP, a differentiable version of PLP estimation that replaces the max operation when selecting a locally optimal periodicity kernel with a softmax-based weighting scheme. This modification ensures good gradient flow, allowing PLP to be seamlessly integrated into deep learning pipelines as an intermediate layer or as part of the loss function. We provide technical insights into its differentiable formulation and present experiments comparing it to the original non-differentiable PLP approach. Additionally, case studies in beat tracking highlight the advantages of dPLP in improving periodicity-aware representations within neural network architectures.

1. INTRODUCTION

Rhythm, a fundamental component of music, is shaped by beats (regular pulses), tempo (the rate at which those beats occur), and meter (the grouping of beats into measures). As rhythm involves the organization of elements across multiple hierarchical levels, its analysis remains a challenging task in MIR [1,2]. Predominant Local Pulse (PLP) estimation, designed to analyze and enhance the local periodicity of musical novelty functions [3, 4], serves as an effective tool for rhythm analysis [5–7] and beat tracking [8–10]. Relying on the idea of the Fourier tempogram, the method of PLP analyzes an input novelty function and derives for each time position an optimal sinusoidal kernel that best represents the local peak structure of the novelty function. By overlap-adding these derived sinusoids for all time positions and applying rectification, a PLP function which represents the periodicity enhancement of the original novelty function can be derived. However, the process of determining at each time position the optimal sinusoidal kernel representing predominant periodicity relies on a non-differentiable max operation, restricting PLP's integration with modern deep-learning frameworks. Consequently, existing studies employ PLP as a post-processing technique, isolated from the system's training process. For example, in beat tracking, current neural networks often lack an explicit mechanism to learn and produce periodic outputs, thus depending on a separate post-processor like PLP [8, 9] or a dynamic Bayesian Network (DBN) [11–13], which enforces periodicity through stronger tempo assumptions. This two-stage architecture not only reveals the limitations of what existing neural networks can learn but also necessitates manual adjustments to post-processing settings when their tempo assumptions are violated. [1]

With the growing demand for interpretable, efficient, and controllable models, researchers are increasingly developing differentiable variants of model-based approaches. For instance, by replacing the minimal-cost alignment in dynamic time warping (DTW) with a soft-minimum calculation, Cuturi and Blondel [14] introduced soft-DTW, enabling its use as a differentiable loss function for training neural networks on weakly aligned data [15, 16]. Similarly, differentiable digital signal processing (DDSP) methods [17–21] have emerged following this trend. Building on these advancements and addressing existing limitations, we introduce dPLP, a differentiable variant of PLP estimation. Designed for seamless integration into deep learning pipelines, dPLP replaces the non-differentiable max operation with a softmax-based weighting scheme, enabling smooth optimization. To evaluate its benefits, we conduct a proof-of-concept beat tracking experiment on a small dataset of popular music. We introduce a lightweight, differentiable spectral flux variant as a trainable activity estimator. By integrating this module with dPLP, we establish a model-based, interpretable

[1] The DBN, for instance, requires hyperparameters to define a tempo change distribution, affecting the model's flexibility in handling tempo variations. Likewise, the PLP requires a predefined kernel size to estimate local periodicity. If the selected kernel size is too short, it may fail to capture periodicity from the input novelty function; if too long, it may introduce noise by capturing inconsistent periodicity from different regions.

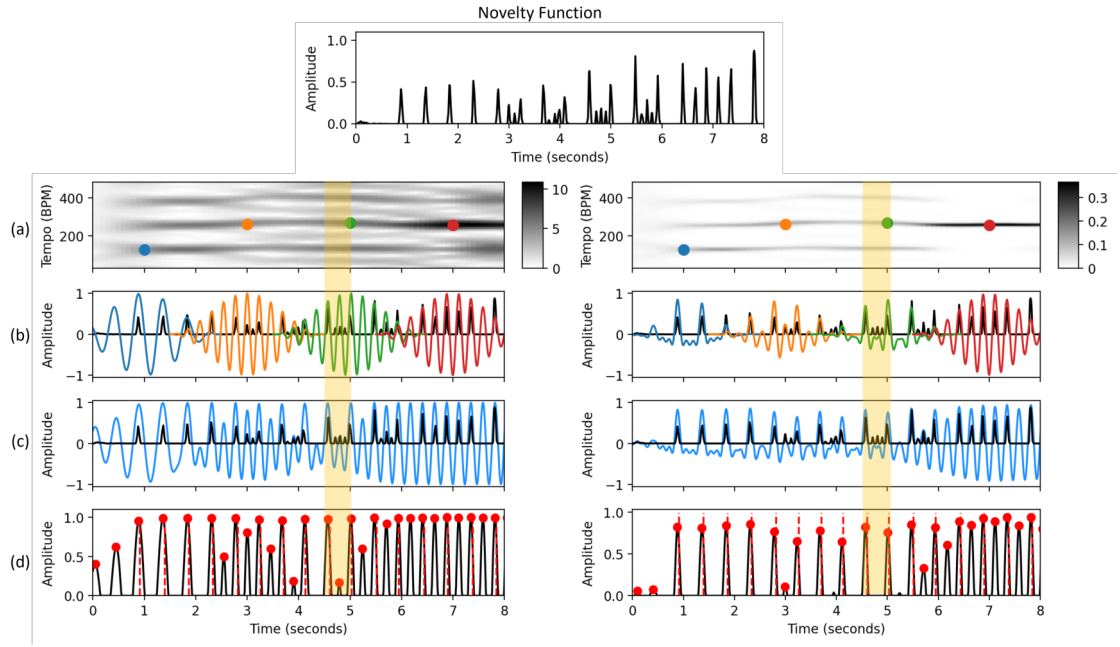

Figure 1. Comparison of the original PLP (left) and dPLP (right) calculation pipelines. (Top) Input novelty function, duplicated in (b) and (c) for reference. **(a)** Fourier magnitude tempogram (left) and its frame-wise softmax-transformed variant (right). **(b)** Optimal (left) vs. weighted-summed (right) sinusoidal kernels at four time positions. **(c)** Kernel accumulation. **(d)** Derived PLP/dPLP functions (black curves) with peak positions identified by a peak picker. Annotated beat positions are marked by vertical red dashed lines. The yellow region highlights differences between PLP and dPLP.

framework that enhances the model's ability to capture periodicity.

The remainder of this paper is structured as follows. Section 2 introduces the formulation, computation, and key parameters of dPLP. Section 3 presents a beat tracking case study, outlining the research questions and baseline architectures. Section 4 analyzes the experimental results, providing both quantitative and qualitative evaluations. Finally, Section 5 concludes the study.

2. MATHEMATICAL FORMULATION OF dPLP

In this section, we introduce the mathematical notation and formulas for both the classical PLP and differentiable PLP.

2.1 Original PLP

Figure 1 (left) illustrates the computation of the original PLP function [3]. Given a novelty function $\Delta : \mathbb{Z} \to \mathbb{R}$ (Figure 1, top), representing the onset envelope or beat likelihood, PLP estimates a periodicity-enhanced version of Δ (Figure 1d, left). The process applies a discrete STFT to Δ using a window function $\mathcal{W} : \mathbb{Z} \to \mathbb{R}$. This window, for example a Hann window, is of length $K \in \mathbb{N}$, centered at $n = 0$, and zero outside. For frequency $\omega \in \mathbb{R}_{\geq 0}$ and time $n \in \mathbb{Z}$, the Fourier coefficient $\mathcal{F}(n, \omega)$ is defined as

$$\mathcal{F}(n, \omega) = \sum_{m \in \mathbb{Z}} \Delta(m) \mathcal{W}(m - n) e^{-2\pi i \omega m}. \quad (1)$$

Let $\Theta \subset \mathbb{R}_{>0}$ be a finite set of tempi, specified in beats per minute (BPM). The discrete Fourier tempogram $\mathcal{T} : \mathbb{Z} \times \Theta \to \mathbb{R}_{\geq 0}$ is defined as the magnitude of the Fourier coefficient, given by

$$\mathcal{T}(n, \tau) = |\mathcal{F}(n, \tau/60)|. \quad (2)$$

Let $\varphi(n, \tau)$ denote the phase of $\mathcal{F}(n, \tau/60)$. The corresponding windowed sinusoidal kernel at time n with tempo τ is

$$\kappa_{n,\tau}(m) := \mathcal{W}(m - n) \cos\Big(2\pi\big(m \cdot \tau/60 - \varphi(n, \tau)\big)\Big), \quad (3)$$

where $\mathcal{W}$ is the same window function as in the STFT computation. The original (argmax-based) PLP estimation finds for each n the tempo $\tau_n \in \Theta$ that maximizes the magnitude tempogram $\mathcal{T}(n, \tau)$ (Figure 1a, left):

$$\tau_n := \operatorname*{argmax}_{\tau \in \Theta} \mathcal{T}(n, \tau). \quad (4)$$

Using τ_n and the corresponding phase $\varphi(n, \tau_n)$, the optimal sinusoidal kernel $\kappa_{n,\tau_n}(m)$ (Figure 1b, left) can be derived by Equation 3. The derived sinusoids are accumulated over time by overlap-adding (Figure 1c, left), preserving periodicity while allowing local tempo variations. Finally, half-wave rectification (omitting negative values) yields the original PLP function $\Gamma : \mathbb{Z} \to \mathbb{R}_{\geq 0}$ (Figure 1d, left):

$$\Gamma(m) = \left| \sum_{n \in \mathbb{Z}} \kappa_{n,\tau_n}(m) \right|_{\geq 0}. \quad (5)$$

In summary, PLP estimation employs Fourier coefficients and the Fourier tempogram to extract local sinusoidal kernels that model periodicity. Overlap-adding these kernels reconstructs a periodicity-enhanced function.

The resulting PLP function depends on Δ's quality, the window size K, and the tempo set Θ, requiring careful parameter selection.

2.2 Differentiable PLP

One step that makes the PLP computation non-differentiable is the argmax operation in Equation 4, only retaining the windowed sinusoid that fits best. To obtain a soft and differentiable approximation of the optimal windowed sinusoid, we instead apply the softmax function, replacing the optimal windowed sinusoid in Equation 3 by a weighted sum of all windowed sinusoids.

To this end, we compute weight factors for all windowed sinusoids using the softmax function

$$\sigma_n^\gamma(\tau) := \frac{\exp\left(\mathcal{T}(n,\tau)/\gamma\right)}{\sum_{\tau'\in\Theta}\exp\left(\mathcal{T}(n,\tau')/\gamma\right)}, \tag{6}$$

where $\gamma > 0$ is a temperature hyperparameter that controls the softness of the distribution.

For $\gamma \to 0$, $\sigma_n^\gamma(\tau)$ approximates the argmax operation, meaning the largest value of $\mathcal{T}(n,\tau)$ dominates, resulting in a one-hot distribution. Conversely, for large γ, the softmax output becomes more uniform, with all values of $\sigma_n^\gamma(\tau)$ tending toward $1/|\Theta|$.

Using these weights, we compute a soft approximation of the optimal windowed sinusoid as

$$\kappa_n^\gamma(m) := \sum_{\tau\in\Theta}\sigma_n^\gamma(\tau)\cdot\kappa_{n,\tau}(m), \tag{7}$$

where $\sigma_n^\gamma(\tau)$ is the softmax output, representing weights for all windowed sinusoids.

Since the softmax function is differentiable, κ_n^γ is differentiable with respect to $\mathcal{T}(n,\tau)$. The dPLP function is then:

$$\Gamma^\gamma(m) = \left|\sum_{n\in\mathbb{Z}}\kappa_n^\gamma(m)\right|_{\geq 0}. \tag{8}$$

Figure 1 (right) illustrates the dPLP computation. Given the softmax-normalized tempogram (Figure 1a, right), the weighted-summed sinusoidal kernel at time n is a weighted sum of the kernels of all tempi. Constructive or destructive interference modifies kernel shapes compared to the argmax case (Figure 1b, left). For time positions with ambiguous tempo (e.g., orange and green dots), κ_n^γ preserves fewer peaks due to destructive interference. For positions with a dominant tempo (e.g., red dot), the softmax and argmax kernels are nearly identical. As shown in Figure 1d (yellow regions), these kernel differences affect beat estimates when PLP/dPLP functions serve as beat novelty functions. Overall, dPLP behaves as an intermediary between the original novelty function and the original PLP, offering a differentiable module for periodicity enhancement, with the softness adjustable via the softmax temperature parameter γ.

2.3 Hyperparameters

As indicated in Sections 2.1 and 2.2, the properties of the original PLP and dPLP depend largely on the hyperparameters of the window (kernel) length K and the tempo range Θ.[2] In this study, we experiment with kernel sizes of 3, 5, and 10 seconds—corresponding to $K \in \{300, 500, 1000\}$ frames at a frame rate of 100 Hz—to cover varying lengths of local temporal context. For Θ, we consider tempi ranging from 20 to 320 BPM, using two types of scales: linear (`LN`) and logarithmic (`LG`). In the `LN` scale, Θ is defined as $\{\tau \in \mathbb{N} \mid 20 \leq \tau \leq 320\}$, resulting in a total of 301 tempo classes. In the `LG` scale, Θ consists of 81 values, also ranging from 20 to 320 BPM, spaced evenly on a logarithmic scale. Since humans are sensitive to relative changes in tempo rather than absolute differences [22], the `LG` scale aligns better with human perception. It achieves good coverage of the tempo range with fewer tempo classes than the `LN` scale, reducing the computational cost of tempogram and dPLP computation.

Additionally, since our focus is to explore the properties and potential benefits of incorporating dPLP rather than optimizing hyperparameters for a specific case, we fix the softmax temperature at $\gamma = 1$ in this study. The effectiveness of these choice is evaluated in Section 3.

3. CASE STUDY IN BEAT TRACKING

We conduct a case study on beat tracking, following the conventional architecture, which consist of an activity estimator and a post-processor [23, 24]. The activity estimator converts audio features (e.g., spectrograms) into real-valued novelty curves, indicating the likelihood of each time frame containing a beat. The post-processor then refines these curves into final binary beat estimates. This experiment aims to illustrate the advantages of the dPLP method, which enables backpropagation. Rather than advancing the state of the art in beat tracking, it serves as a controlled demonstration. To ensure efficient training and controlled analysis, we use a small toy dataset (Section 3.1), keep all network components minimal, integrate dPLP in various ways (Section 3.2), and employ a peak-picking-based post-processing method (Section 3.3). The resulting beat estimates are evaluated in Section 3.4 to assess dPLP's impact and functionality.

3.1 Datasets

The GTZAN dataset [25, 26] is a widely used benchmark for music genre classification and various audio analysis tasks, including beat tracking [13, 24, 27, 28]. It comprises 1,000 audio tracks, each 30 seconds long, spanning ten genres, offering a diverse collection of musical styles. In this study, we specifically focus on the 100 tracks of popular music, providing a simplified scenario to examine the

[2] Note that when calculating the Fourier tempogram and the corresponding PLP function, the hop size is also a hyperparameter that affects temporal resolution and computational cost. For simplicity, we empirically fix the hop size to 10 frames without further discussion.

behaviors and effects of the proposed ideas. For the following beat tracking experiments, we randomly split these 100 tracks into 60 for training, 20 for validation, and 20 for testing, reporting results for the test data.

3.2 Beat Activity Estimators

Given a 44.1 kHz audio recording, we compute STFT-based spectrograms using `librosa` [29] with an FFT size of 2048, a window length of 1024, and a hop size of 441, resulting in spectrograms with a 100 Hz temporal resolution. As shown in Figure 2, these spectrograms serve as the primary input feature for subsequent experiments.

3.2.1 Spectral Flux

Spectral flux [3, 4, 30] is a widely used model-based technique for onset detection. Given an STFT spectrogram, it applies logarithmic compression, discrete differentiation, half-wave rectification, and accumulation to generate a novelty curve. To further refine its quality, baseline subtraction, Gaussian smoothing, and normalization are often incorporated. To evaluate the impact of dPLP's differentiability, we use bandwise spectral flux as an example for novelty function computation. As a minor contribution of this study, we implement a lightweight, trainable version of spectral flux (`SFX`) by formulating it as a `PyTorch` module. The `SFX` module processes an STFT spectrogram by dividing it into eight frequency bands, computing spectral flux independently for each band [3, 4, 30], applying a weighted sum, and performing Gaussian smoothing, rectification, and max-normalization. To introduce learnability, we make the differentiation convolution kernels, log compression parameters, and bandwise weighting parameters trainable, resulting in a total of 64 trainable parameters.

We train `SFX` for beat tracking using 60 training tracks from GTZAN (Section 3.1), with a batch size of 8, a learning rate of 0.1, the Adam optimizer, and weighted binary cross-entropy (BCE) loss. [3] The module is initialized with first-order differentiation convolution kernels, a log compression parameter of 10, and average-weighted summation, referred to as `SFX-I`. After training, the resulting model is denoted as `SFX-T`.

3.2.2 Argmax PLP and Softmax PLP

To compare the properties and behavior of the original argmax-based PLP (`A-`$*$) and the proposed differentiable softmax-based PLP (`S-`$*$), we process the beat novelty generated by the trained `SFX-T` using both methods separately. Following Section 2.3, we experiment with three PLP kernel sizes K (3, 5, and 10 seconds) and two tempo scales: linear (`LN`) and logarithmic (`LG`). The resulting PLP functions are peak-picked (Section 3.3) and evaluated (Section 3.4) for comparison (Section 4.1).

3.2.3 dPLP Incorporated Architecture

Figure 2 illustrates our proposed dPLP-incorporated architecture, consisting of a spectral flux module (`S`, identical to `SFX`), a dPLP module, and a fuser (`F`). This design integrates an onset-based activity estimator (`S`), a periodicity analyzer (dPLP), and a fuser (`F`) that learns to combine information from both components. Given an STFT spectrogram, the `S` module generates a beat novelty function Δ_{S}. The dPLP module processes Δ_{S} with three kernel sizes ($K \in \{3, 5, 10\}$ seconds), producing three dPLP curves. The fuser (`F`) concatenates Δ_{S} with these dPLP curves ($\Gamma^{\gamma} - K*$), applies weighted summation and smoothing, and outputs the final beat novelty function Δ_{F}. The fuser (`F`) comprises a linear layer, a convolutional layer, and a sigmoid activation, totaling 21 trainable parameters.

We refer to this architecture, where both `S` and `F` are trainable, as `M1`. During training, the beat novelty function Δ_{F} generated by the fuser (denoted as `M1-F`) is compared with reference beat annotations, and the BCE loss guides the learning process. Since dPLP enables gradient backpropagation from `M1-F` through the dPLP to the `S` module (`M1-S`), `M1-S` is optimized by a combined loss function incorporating dPLP, the fuser, and BCE loss. We are particularly interested in whether `M1-S` behaves differently from the standalone-trained `SFX-T`.

To assess the complementarity between Δ_{S} and the dPLP curves ($\Gamma^{\gamma} - K*$), we modify region A (see Figure 2) and implement an ablation model, `M2`. In `M2`, the `S` module (denoted as `M2-S`) is initialized with `SFX-T` parameters and kept fixed, allowing only the fuser (`M2-F`) to be trained. This setup evaluates whether `M2-F` can effectively integrate information from `M2-S` and dPLP

Finally, to evaluate the dPLP module's ability to provide periodicity-based information and enhance beat tracking, we introduce `M3` by modifying region B (also shown in Figure 2). In `M3`, the three dPLP curves are replaced with three duplicated `M3-S` beat novelty functions Δ_{S}, keeping the model size identical to `M1`. Since `M1` and `M3` share the same trainable components (`S` and `F`) and model size, their only difference—presence or absence of dPLP—allows us to directly assess dPLP's effectiveness.

3.3 Peak-Picking-based Post-processing

For all the derived beat novelty functions, we follow [4] and apply one-dimensional Gaussian smoothing (with $\sigma = 3$ frames), max-normalization, and local average-based peak picking (using a 20-second averaging window) to obtain the beat estimations. Compared to conventional post-processors such as DBN [11, 24, 31], this peak picking method avoids strong tempo assumptions and better reflects the properties of the novelty functions, as [10,13,32].

3.4 Evaluation

We evaluate beat estimates using the F1-score (F1), precision (P), and recall (R) as implemented in `mir_eval` [33], with a tolerance window of ± 70 ms. Additionally, since PLP is incorporated to enhance the model's ability to handle longer musical context, we report the L-correct metric [3]. This metric requires at least L consecutive reference beats to be correctly detected rather than considering reference beats individually. For

[3] To address class imbalance, we assign a weight of 3 to the beat class in the BCE loss, as non-beat frames dominate.

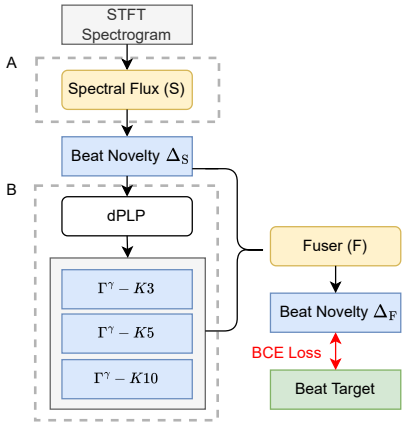

Figure 2. The proposed architecture (M1) and the regions (dotted squares) to modify for the two ablations (M2, M3).

simplicity, we use $L = 2$ and report F-measure (F-L2), precision (P-L2), and recall (R-L2).

4. EXPERIMENT RESULTS

In the following, we analyze the beat tracking results both quantitatively and qualitatively, based on beat estimates derived from the aforementioned novelty functions (Section 3.2) and a simple peak-picking method (Section 3.3).

4.1 Comparison of PLP Settings

Table 1 (top) presents the F-measure and L-correct evaluation results for the standalone spectral flux module (SFX-⋆), comparing the original argmax PLP (A-⋆) with the differentiable softmax PLP (S-⋆). The results indicate that SFX-⋆ performs as expected. As an onset-based module, SFX-I achieves high recall (0.950) but low precision (0.311), resulting in an unsatisfactory F1-score (0.464) and low L-correct values (all below 0.100). After training, SFX-T improves the precision-recall balance, increasing precision from 0.311 to 0.436 and decreasing recall from 0.950 to 0.888. This leads to a higher F1-score (0.576) and improved L-correct values (all above 0.100). Using the SFX-T beat novelty as input, the derived argmax PLP functions (A-⋆) and softmax PLP functions (S-⋆) generally improve beat tracking performance.[4] Specifically, since the test tracks consist of popular music with stable tempi, the dPLP functions help filter out non-beat onsets that do not align with the locally detected periodicity while also enhancing weak onsets at beat positions. This results in improved precision (all above 0.520) and recall (all above 0.900) compared to SFX-T. Moreover, the similar F1-scores (around 0.664) indicate that there is little difference between the linear tempo scale (LN) and the logarithmic tempo scale (LG), as well as between the argmax PLP (A) and the softmax PLP (S). Therefore, we use LG, which is computationally more efficient, for the subsequent experiments involving dPLP-incorporated architectures.

[4] For each setting (e.g., A-LG), we apply three kernel settings ($K \in \{3, 5, 10\}$ seconds), derive three sets of PLP curves, evaluate them separately, and report the averaged scores in Table 1.

Act.	F-Measure			L-Correct		
	F1	P	R	F-L2	P-L2	R-L2
SFX-I	0.464	0.311	**0.950**	0.031	0.022	0.055
SFX-T	**0.576**	**0.436**	0.888	**0.153**	**0.122**	**0.213**
A-LG	0.663	0.531	0.925	0.169	0.152	0.196
A-LN	0.662	0.528	0.929	0.158	0.142	0.183
S-LG	**0.671**	**0.550**	0.908	**0.196**	**0.182**	**0.220**
S-LN	0.664	0.528	**0.933**	0.165	0.147	0.193
M1-F	**0.707**	**0.660**	0.809	**0.470**	**0.445**	**0.519**
M2-F	0.684	0.615	0.817	0.385	0.360	0.439
M3-F	0.664	0.576	**0.819**	0.417	0.375	0.489
M1-S	0.561	0.412	0.921	**0.182**	**0.140**	**0.267**
M2-S	**0.576**	**0.436**	0.888	0.153	0.122	0.213
M3-S	0.529	0.371	**0.952**	0.083	0.060	0.138

Table 1. Beat tracking results. A and S denote argmax and softmax PLP. LG and LN indicate log-scale and linear-scale tempo spaces. F and S represent the fuser and spectral flux module of the dPLP incorporated architecture in Figure 2.

4.2 Effectiveness of Differentiability

Table 1 (middle) presents the results for the fusers (F) of the dPLP-incorporated architectures (M1, M2, M3). Notably, compared to other fusers and baselines, M1-F achieves substantial improvements across all evaluation metrics except recall. Specifically, its superior F1-score (0.707), precision (0.660), and F-L2 (0.470) suggest that M1-F has learned a more effective beat-tracking mechanism. Alternatively, the observed improvements may indicate that the proposed architecture (M1) has a greater capacity to fit the relatively simple structure of popular music in GTZAN. The results from the ablation models further support this observation. Comparing M2-F with SFX-T and the softmax PLPs (S-LG), we find clear evidence of complementarity between the M2 dPLP curves and the M2-S beat novelty function, which M2-F effectively leverages. Specifically, M2-F aligns more closely with the consensus across all input curves, significantly improving precision (from below 0.550 to 0.615) and L-correct metrics (from below 0.220 to above 0.360). However, since M2-S is fixed and non-trainable, M2-F does not benefit from the differentiability of dPLP, resulting in lower capacity compared to M1-F.

The results from M3-F are also noteworthy. Compared to SFX-T, M3-F, which has a larger model size but lacks a dPLP module, adopts a different precision-recall trade-off: it suppresses peaks from M3-S, leading to higher precision (0.576 vs. 0.436), lower recall (0.819 vs. 0.888), and significantly improved L-correct metrics (above 0.370 vs. below 0.220). This suggests that the performance gains observed in M1-F, M2-F, and M3-F over SFX-T may also be influenced by model size. Finally, when comparing M3-F to M1-F, the slightly lower F1-scores of M3-F suggest that, without the dPLP module, M3-F may have lower capacity than both M1-F and M2-F.[5]

[5] At first glance, it may seem contradictory that M3-F achieves higher L-correct metrics than M2-F despite having lower precision (0.576 vs. 0.615). This can be attributed to two factors: (1) The lower L-correct of

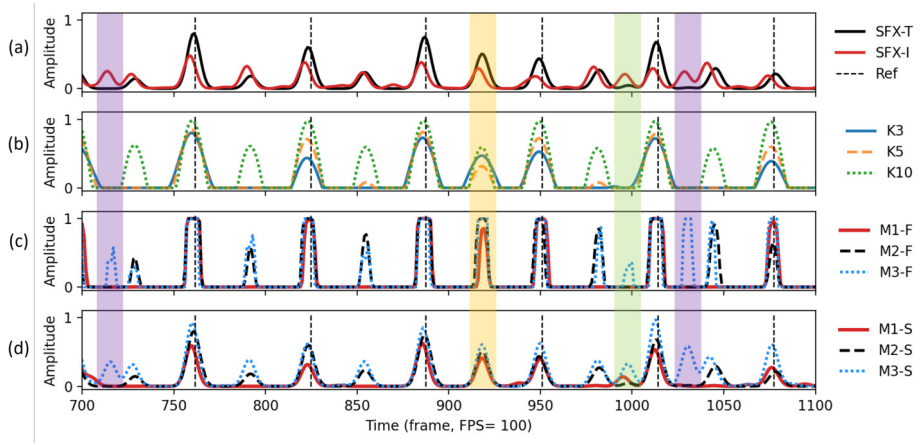

Figure 3. Novelty functions. **(a)** Beat novelty functions from the spectral flux modules (`SFX-*`). **(b)** dPLP curves from `M1`, where K* denotes the dPLP function computed with a kernel size of * seconds. **(c)** Beat novelty functions from fusers (`F`). **(d)** Beat novelty functions from the spectral flux heads (`S`). Black dashed lines indicate annotated reference beats.

Lastly, as shown in Table 1 (bottom), spectral flux modules trained with different loss functions exhibit distinct behaviors, with F1-score differences ranging from 0.03 to 0.05 and variations in other metrics between 0.02 and 0.10.

4.3 Comparison of the Novelty Functions

Figure 3 compares the output novelty functions calculated from a test track in the GTZAN dataset, summarizing our previous discussions. In Figure 3a, the alignment between the reference annotated beats (black vertical dashed lines) and the `SFX-*` novelty functions confirms the high recall and low precision of `SFX-*`, as observed in Table 1. Moreover, compared to `SFX-I`, the trained `SFX-T` learns to suppress several non-beat peaks (purple regions).

Figure 3b shows that `M1` dPLP functions computed with different kernel sizes (K) exhibit varying peak distributions, yet they largely agree at beat positions. Figure 3c visualizes the distinct behaviors of the three fusers (`M1-F`, `M2-F`, `M3-F`). Notably, each fuser exhibits different false-positive errors (e.g., purple or green regions). These differences can be attributed to the presence of the dPLP module (`M1` and `M2` vs. `M3`) and to whether module `S` is further optimized using the gradients backpropagated through the dPLP module (`M1` vs. `M2`). Specifically, the false-positive error shared by all fusers around the 920th frame (yellow region) reveals that all fusers attempt to produce peaks at positions where the input novelty functions agree. In contrast, when the novelty functions from the `S` modules (`M*-S`) do not align with the dPLP curves (e.g., green region), `M1-F` and `M2-F`, which have access to the dPLP outputs, avoid making a false-positive error. Lastly, Figure 3d illustrates the different behaviors of the `S` modules when supervised by different loss functions. Specifically, since the `M3` architecture lacks periodicity information, the `M3-S` head is trained to be more sensitive, generating more and stronger false-positive peaks at non-beat positions (purple regions) compared to `M1-S` and `M2-S`. In contrast, with the additional benefit of gradient backpropagation through dPLP, `M1-S` behaves differently from `M2-S` and `M3-S`, suppressing many non-beat onsets (e.g., around the 730th, 790th, and 860th frames).

5. CONCLUSION

In this paper, we presented a differentiable variant of Predominant Local Pulse (dPLP) estimation, replacing the non-differentiable selection of an optimal windowed sinusoid with a softmax-based weighted summation. While dPLP behaves similarly to the original algorithm in terms of enhancing periodicity in the input signal, its differentiability enables seamless integration into deep learning pipelines and supports end-to-end training.

The main contribution of this work lies on a conceptual level—namely, in the formulation of dPLP as an interpretable, flexible, and differentiable module for periodicity enhancement. To illustrate the behavior and potential benefits of dPLP in a controlled setting, we conducted a proof-of-concept experiment on beat tracking. As part of this setup, we also introduced a lightweight differentiable variant of the spectral flux method, which serves as a simple but trainable activity estimator. While this differentiable spectral flux is a minor contribution, it demonstrates how model-based components can be incorporated into learning frameworks.

Our experimental results highlight the potential of combining differentiable modules like dPLP with trainable feature extractors in an end-to-end fashion. In future work, we plan to integrate dPLP into more advanced architectures and further investigate its interaction with other system components. Overall, we believe that dPLP can serve as a valuable building block for improving the transparency, controllability, and interpretability of rhythm analysis and beat tracking systems.

`M2-F` is partly due to dPLP's bias toward faster tempi, which can cause taps to align with tempo harmonics (e.g., double tempo), disrupting the continuity required by L-correct. (2) The lower precision of `M3-F` results from non-beat onsets clustering around specific beats, introducing false positives. Unlike the evenly distributed octave errors in `M2-F`, these false positives are more localized, allowing `M3-F` to achieve higher L-correct values.

6. ACKNOWLEDGEMENTS

This work was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Grant No. 500643750 (MU 2686/15-1). The International Audio Laboratories Erlangen are a joint institution of the Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU) and the Fraunhofer Institute for Integrated Circuits IIS.

7. REFERENCES

[1] P. Grosche, M. Müller, and F. Kurth, "Cyclic tempogram – a mid-level tempo representation for music signals," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Dallas, Texas, USA, 2010, pp. 5522–5525.

[2] G. T. Toussaint, "The geometry of musical rhythm," in *Proceedings of the Japanese Conference on Discrete and Computational Geometry (JCDCG)*, Tokyo, Japan, 2004, pp. 198–212.

[3] P. Grosche and M. Müller, "Extracting predominant local pulse information from music recordings," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 6, pp. 1688–1701, 2011.

[4] M. Müller and C.-Y. Chiu, "A basic tutorial on novelty and activation functions for music signal processing," *Transactions of the International Society for Music Information Retrieval (TISMIR)*, vol. 7, no. 1, pp. 179–194, 2024.

[5] P. Grosche and M. Müller, "A mid-level representation for capturing dominant tempo and pulse information in music recordings," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Kobe, Japan, Oct. 2009, pp. 189–194.

[6] S. P. Bhatta, S. Nagaraj Bharadwaj, S. Shadakshari, and A. Bhat, "Laya estimation for Hindustani classical vocals, devoid of rhythmic indicators," in *Proceedings of the International Conference on Electronics, Computing and Communication Technologies (CONECCT)*, Bangalore, India, 2024.

[7] P. Meier, S. Schwär, and M. Müller, "A real-time approach for estimating pulse tracking parameters for beat-synchronous audio effects," in *Proceedings of the International Conference on Digital Audio Effects (DAFx)*, Guildford, Surrey, UK, 2024, pp. 314–321.

[8] P. Grosche, M. Müller, and C. S. Sapp, "What makes beat tracking difficult? A case study on Chopin Mazurkas," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Utrecht, The Netherlands, 2010, pp. 649–654.

[9] P. Meier, C.-Y. Chiu, and M. Müller, "A real-time beat tracking system with zero latency and enhanced controllability," *Transactions of the International Society for Music Information Retrieval (TISMIR)*, vol. 7, no. 1, pp. 213–227, 2024.

[10] C.-Y. Chiu, M. Müller, M. E. P. Davies, A. W.-Y. Su, and Y.-H. Yang, "Local periodicity-based beat tracking for expressive classical piano music," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 31, pp. 2824–2835, 2023.

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

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

[13] F. Foscarin, J. Schlüter, and G. Widmer, "Beat this! Accurate beat tracking without DBN postprocessing," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, San Francisco, CA, United States, 2024, pp. 962–969.

[14] M. Cuturi and M. Blondel, "Soft-DTW: a differentiable loss function for time-series," in *Proceedings of the International Conference on Machine Learning (ICML)*, Sydney, NSW, Australia, 2017, pp. 894–903.

[15] M. Krause, C. Weiß, and M. Müller, "Soft dynamic time warping for multi-pitch estimation and beyond," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Rhodes Island, Greece, 2023.

[16] J. Zeitler, S. Deniffel, M. Krause, and M. Müller, "Stabilizing training with soft dynamic time warping: A case study for pitch class estimation with weakly aligned targets," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2023, pp. 433–439.

[17] J. Engel, R. Swavely, L. H. Hantrakul, A. Roberts, and C. Hawthorne, "Self-supervised pitch detection by inverse audio synthesis," in *International Conference on Machine Learning (ICML), Workshop on Self-Supervision in Audio and Speech*, Vienna, Austria, 2020.

[18] J. Engel, L. Hantrakul, C. Gu, and A. Roberts, "DDSP: Differentiable digital signal processing," in *Proceedings of the International Conference on Learning Representations (ICLR)*, Virtual, 2020.

[19] Y. Yang, M. Hira, Z. Ni, A. Astafurov, C. Chen, C. Puhrsch, D. Pollack, D. Genzel, D. Greenberg, E. Z. Yang, J. Lian, J. Hwang, J. Chen, P. Goldsborough, S. Narenthiran, S. Watanabe, S. Chintala, and

V. Quenneville-Bélair, "Torchaudio: Building blocks for audio and speech processing," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Virtual and Singapore, 2022, pp. 6982–6986.

[20] M. Leiber, Y. Marnissi, A. Barrau, and M. E. Badaoui, "Differentiable adaptive short-time fourier transform with respect to the window length," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Rhodes Island, Greece, 2023, pp. 1–5.

[21] ——, "Differentiable short-time fourier transform with respect to the hop length," in *IEEE Statistical Signal Processing Workshop (SSP)*, Hanoi, Vietnam, 2023, pp. 230–234.

[22] K. Thomas, "Just noticeable difference and tempo change," *Journal of Scientific Psychology*, vol. 2, pp. 14–20, 2007.

[23] 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 *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Paris, France, 2018, pp. 106–112.

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

[25] G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," *IEEE Transactions on Speech and Audio Processing*, vol. 10, no. 5, pp. 293–302, 2002.

[26] U. Marchand and G. Peeters, "Swing ratio estimation," in *Proceedings of the International Conference on Digital Audio Effects (DAFx)*, Trondheim, Norway, 2015, pp. 423–428.

[27] D. Desblancs, V. Lostanlen, and R. Hennequin, "Zero-note samba: Self-supervised beat tracking," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 31, pp. 2922–2934, 2023.

[28] Y. Hung, J. Wang, X. Song, W. T. Lu, and M. Won, "Modeling beats and downbeats with a time-frequency transformer," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Virtual and Singapore, 2022, pp. 401–405.

[29] B. McFee, C. Raffel, D. Liang, D. P. Ellis, M. McVicar, E. Battenberg, and O. Nieto, "Librosa: Audio and music signal analysis in Python," in *Proceedings the Python Science Conference*, Austin, Texas, USA, 2015, pp. 18–25.

[30] M. Müller and F. Zalkow, "libfmp: A Python package for fundamentals of music processing," *Journal of Open Source Software (JOSS)*, vol. 6, no. 63, pp. 3326:1–5, 2021.

[31] S. Böck, F. Korzeniowski, J. Schlüter, F. Krebs, and G. Widmer, "madmom: A new Python audio and music signal processing library," in *Proceedings of the ACM International Conference on Multimedia (ACM-MM)*, Amsterdam, The Netherlands, 2016, pp. 1174–1178.

[32] C.-Y. Chiu, L. Liu, C. Weiß, and M. Müller, "Cross-modal approaches to beat tracking: A case study on Chopin Mazurkas," *Transaction of the International Society for Music Information Retrieval (TISMIR)*, vol. 8, no. 1, pp. 55–69, 2025.

[33] C. Raffel, B. McFee, E. J. Humphrey, J. Salamon, O. Nieto, D. Liang, and D. P. W. Ellis, "MIR_EVAL: A transparent implementation of common MIR metrics," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Taipei, Taiwan, 2014, pp. 367–372.