BeatMamba: A Drum-Separated Beat Tracking Method Using Mamba

1st Xiang Li
School of Information and Intelligent Science, Donghua University
Shanghai, China
llllxx2024@163.com

2nd Xiaoliang He
School of Information and Intelligent Science, Donghua University
Shanghai, China
2232988@mail.dhu.edu.cn

3rd Xiaopeng Zhao*
School of Information and Intelligent Science, Donghua University
Shanghai, China
zxp@dhu.edu.cn

***Abstract*—In this paper, we propose BeatMamba, a drum-separated beat tracking framework built on the Mamba state space architecture. Beat tracking in polyphonic music is challenged by complex mixtures and high computational costs. To address these issues, We first apply music source separation to isolate the drum stem, preserving salient percussive cues while suppressing interference from vocals and harmony. This representation improves rhythmic clarity and reduces ambiguity in mix audios. For sequence modeling, we employ selective state space modules to capture long-range temporal dependencies with linear-time and memory complexity, avoiding the quadratic cost of self-attention. Compared to Transformer baseline models, BeatMamba significantly improved computational efficiency and reduced model size, making it more suitable for low-latency and real-time applications. Experimental results on benchmark datasets demonstrate that BeatMamba reduces parameter counts while maintaining or even surpassing the accuracy of heavy models[1].**

***Index Terms*—Beat Tracking, Mamba, Music Source Separation, Music Information Retrieval**

I. Introduction

Beat Tracking is the task of determining the temporal position of beats in audio signals. It has been a fundamental challenge in the field of music information retrieval. As the basic temporal component of musical structure, beat serves as the natural synchronising anchor for the listener's perception, forming the rhythmic foundation of musical experience [1]. Robust beat tracking algorithms are essential enablers for downstream MIR applications. They provide accurate time alignment and rhythmic structure modeling for music transcription [2] [3], beat-synchronous temporal cues for music structure analysis [4] [5], and beat-consistent temporal scaffolding in music therapy [6]. In addition, beat tracking has extensive practical applications, providing low-latency and robustness in real-time performance processing and automatic accompaniment for vocal input [7].

In recent years, with the rapid advancement of deep learning, beat tracking tasks have witnessed breakthrough progress. Contemporary neural architectures, including Convolutional Neural Networks (CNN) [8] [9], Recurrent Neural Networks (RNN) [10] [11] [12], have demonstrated substantial improvements on benchmark datasets through their capacity for hierarchical feature extraction and time-frequency pattern recognition. But these methods struggle with long-range dependencies and limit ability for complex rhythmic patterns. Currently, Transformer-based models such as [13] [14] [15] leverage self-attention mechanisms to capture long-term temporal dependencies, solving problems in previous neural networks. However, while these models have achieved state-of-the-art results, their high computational complexity and memory requirements limit real-time processing capabilities. Recently, State Space Models (SSM) [16] [17] have emerged as a promising alternative for sequence modeling, offering linear complexity while maintaining the ability to capture long-range dependencies. Inspired by Mamba architecture [16] [17], we introduce BeatMamba, which incorporates selective state space mechanisms for beat tracking.

On the other hand, beat tracking relies upon identifying periodic patterns within complex audio signals. However, traditional single-channel approaches face challenges in isolating rhythm-related information from intricate musical content, which may ignore beat-related information [18]. To address this limitation, we propose a dual-channel feature extraction strategy that maximizes information through music source separation [19]. Drawing inspiration from the human beat perception mechanism, where listeners automatically focus attention on percussive elements such as drum beats during musical perception [20], we perform isolation operations exclusively on the drum channel, providing complementary rhythmic information and richer representations for subsequent modeling.

Experimental results indicate that our model achieves modeling capabilities comparable to or even surpassing Transformers while using fewer parameters. This breakthrough development paves the way for modeling long audio sequences. The principal contributions of our paper are as follows: 1) We propose a beat tracking method based on Mamba, which requires fewer parameters while maintaining high performance; 2) We employ music source separation to construct a dual-channel feature extraction strategy, enhancing representational capacity; 3) We develop a drum-sensing approach that simulates human beat perception capabilities, thereby improving model performance.

* Corresponding author

[1]Code is available at https://github.com/LLiyee/BeatMamba

II. Related Works

Beat tracking has been a fundamental research problem in music information retrieval for decades. Early beat tracking approaches [21] [22], mainly employed traditional signal processing methods, such as onset detection and phase-based methods. With the advancement of deep learning, various neural network architectures have been proposed to address this challenge. Convolutional Neural Networks have demonstrated well capabilities in extracting local rhythmic patterns. Böck [23] pioneered the application of Temporal Convolutional Networks (TCN) for beat detection. Durand [8] proposed an ensemble of convolutional networks to process different feature characteristics. To better capture temporal dependencies, Recurrent Neural Networks have been widely adopted in beat tracking tasks. Böck [24] introduced a bidirectional Long Short-Term Memory network to perform a frame by frame beat classification of the signal. Krebs [10] developed a downbeat detection system using beat-synchronous features with two layer bidirectional GRU networks. Sequently, Transformer-based models have emerged as the dominant paradigm for beat tracking. Zhao [25] proposed dilated self-attention mechanisms to hierarchically model beat structures while reducing computational complexity. Cheng [14] proposed a two-stage method consisting of a low-resolution encoder and a high-resolution decoder. The low-resolution encoder is used to capture global features with more balanced data and the high-resolution decoder is designed to predict beat times. Hung [13] designed SpecTNT, which models both spectral and temporal dimensions of a time-frequency input of audio. The spectral encoder handles the spectral features to pay attention to harmonic components of each frame and the temporal encoder aggregates useful local spectral information.

In addition, many self-supervised learning methods have received extensive attention in recent years. The core idea is to learn meaningful representations from unlabeled data. Gagneré [26] proposed a self-supervised pre-training scheme inspired by Contrastive Predictive Coding to equip Transformer for few-shot beat tracking. Without any ground-truth tempo or beat annotations, using local maxima of a Predominant Local Pulse function as a proxy for tatum positions. Desblancs [27] introduced two fully-convolutional networks in parallel to perceive beats in musical recordings despite having little or no access to beat annotations.

Despite advances in deep learning based beat tracking techniques, existing methods still face certain limitations. Traditional single-channel approaches often struggle to isolate rhythm-related information from complex polyphonic mixtures. High complexity and large parameter counts limit their deployment in real-time and resource-constrained scenarios. To overcome these issues, we propose BeatMamba, which integrates selective state space mechanisms with a drum-separated dual-channel strategy. Our approach focuses on emulating the perceptual capabilities of human hearing and exploring the potential of applying beat tracking in resource-constrained scenarios.

III. Proposed Method

In this section, we first introduce the overall architecture of BeatMamba. Then we will explain the drum-separated mechanism and the feature processing based on Mamba in detail.

A. Overall Architecture

Fig. 1 shows the overall architecture of our Mamba-based beat tracking model. Our model primarily consists of three major components: the Drum-Separated mechanism as the pre-processing stage; Convolutional layers and Mamba blocks as the intermediate processing stage; and Dynamic Bayesian Network as the post-processing stage. Our frame rate is 43.07 fps, and the raw audio signals are initially sampled at 44.1 kHz. The drum-separated mechanism processes the input audio through a series of steps to derive dual-channel features. The intermediate processing stage proceeds these dual-channel features to generate beat activations. CNN serves as a feature extractor that maps dual-channel features to a unified 256-dimensional feature, which is subsequently fed into 9 Mamba blocks for sequence modeling. We also add a regularization branch to predict global tempo via skip connections [23]. Finally, we employ a Dynamic Bayesian Network (DBN) from the madmom package [28], converting beat activations into the temporal positions of beat points.

B. Drum-Separated Beat Tracking

To investigate the impact of drum channels on beat-tracking tasks, we propose a drum-separated dual-channel feature extraction method. This dual-channel representation preserves both the harmonic content in the original mixture and the rhythmic information in the isolated percussion, enabling joint optimization of complementary rhythmic cues during model training.

To be specific, given an input audio signal $\mathbf{x}(t)$, to capture essential rhythmic and harmonic information for beat detection, we apply Hybrid Transformer Demucs [29] to obtain separated audio streams:

$$\big[\mathbf{x}_{\mathrm{mix}}(t),\, \mathbf{x}_{\mathrm{drum}}(t)\big] = \mathrm{Demucs}\big(\mathbf{x}(t)\big) \tag{1}$$

where $\mathbf{x}_{\mathrm{mix}}(t)$ represents the original mixture and $\mathbf{x}_{\mathrm{drum}}(t)$ contains the isolated percussive component. For each audio stream, we compute Log-Mel spectrograms using Short-Time Fourier Transform:

$$\mathbf{S}_k(f,t) = \big|\mathrm{STFT}\big(\mathbf{x}_k(t)\big)\big|^2 \tag{2}$$

where $k \in \{\mathrm{mix},\, \mathrm{drum}\}$, f denotes frequency bins, t is the time frame. The Mel-scale transformation is then applied via the filter bank matrix $\mathbf{M} \in \mathbb{R}^{F\times N}$:

$$\mathbf{X}_k(m,t) = \big(\mathbf{M}\cdot\mathbf{S}_k(:,t)\big)_m \tag{3}$$

where F is the number of Mel frequency bins, N is the number of FFT frequency bins, and m denotes Mel frequency bins. Finally, we apply logarithmic compression to obtain the Log-Mel spectrogram:

$$\mathbf{X}_k(m,t) = \log(\mathbf{X}_k(m,t) + \varepsilon\big) \tag{4}$$

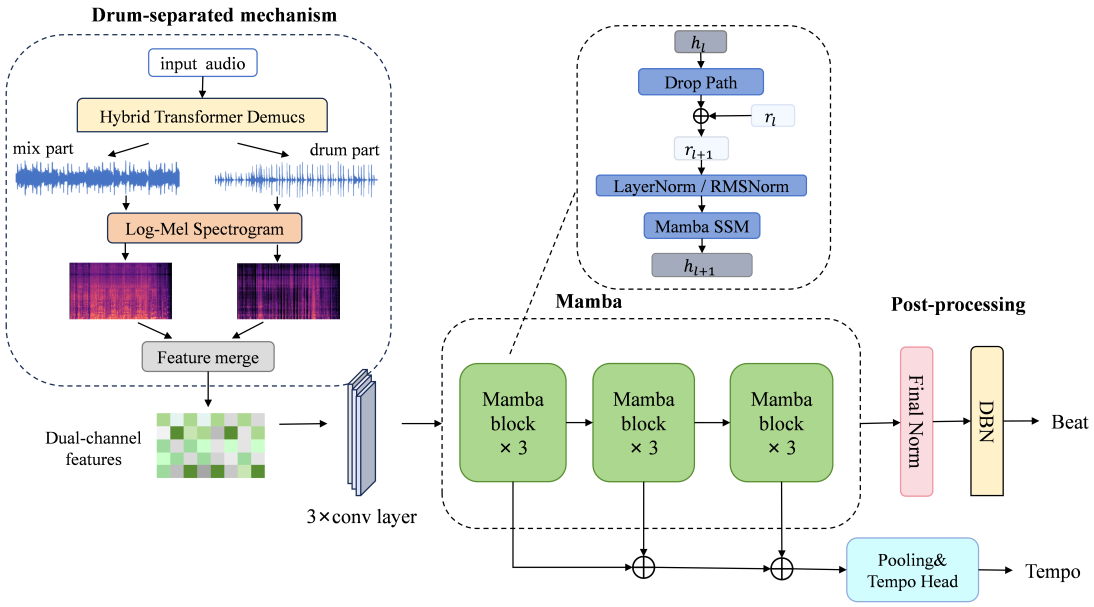

Fig. 1. The overall architecture of BeatMamba

where $\varepsilon = 10^{-8}$ prevents log-domain singularities. The final input features are constructed by channel-wise concatenation:

$$\mathbf{X}(t) = \left[\mathbf{X}_{\text{mix}}(:,t),\ \mathbf{X}_{\text{drum}}(:,t)\right] \in \mathbb{R}^{B\times T\times F\times 2} \tag{5}$$

where $\mathbf{X}_{\text{mix}}$ is the original polyphonic mixture features and $\mathbf{X}_{\text{drum}}$ represents drum-separated features, B is the batch size and T is the temporal length.

By extracting dual-channel features, we can maximize temporal information from the drum, thereby providing richer representations for the subsequent intermediate processing stage.

C. Feature Processing Based on Mamba

We employ a hybrid architecture for intermediate feature processing. The convolutional layers first extract local rhythmic patterns, followed by Mamba blocks that capture long-range dependencies with linear complexity. This design balances local feature extraction with efficient global temporal modeling, avoiding the quadratic complexity and maintaining the ability to capture extended rhythmic structures.

We use three successive convolutional layers to perform forward processing on dual-channel features:

$$\mathbf{H}^{(1)} = \text{MaxPool}(\text{ReLU}(\text{Conv2D}(\mathbf{X}(t), \mathbf{W}^{(1)}))), \tag{6}$$
$$\mathbf{H}^{(2)} = \text{MaxPool}(\text{ReLU}(\text{Conv2D}(\mathbf{H}^{(1)}, \mathbf{W}^{(2)}))), \tag{7}$$
$$\mathbf{H}^{(3)} = \text{MaxPool}(\text{ReLU}(\text{Conv2D}(\mathbf{H}^{(2)}, \mathbf{W}^{(3)}))) \tag{8}$$

Here, $\mathbf{W}^{(i)}$ denotes the convolutional weights for layer i. CNN maps dual-channel features into a unified 256-dimensional feature space. The final CNN output is reshaped to:

$$\mathbf{x} = \text{Reshape}\left(\mathbf{H}^{(3)}\right) \in \mathbb{R}^{B\times T\times d_{\text{model}}} \tag{9}$$

where d_{model} is 256. Next, we process the features from the CNN using a Mamba block. The core Mamba mechanism bases on a selective state space, which is inspired by the continuous system as follows:

$$\begin{aligned} h_t &= \mathbf{A}h_{t-1} + \mathbf{B}x_t \\ y_t &= \mathbf{C}h_t + \mathbf{D}x_t \end{aligned} \tag{10}$$

Here, $h_t \in \mathbb{R}^N$ is the hidden state at time step t, $x_t \in \mathbb{R}^D$ is the input vector, and $y_t \in \mathbb{R}^D$ is the output. The matrices $\mathbf{A} \in \mathbb{R}^{N\times N}$, $\mathbf{B} \in \mathbb{R}^{N\times D}$, $\mathbf{C} \in \mathbb{R}^{D\times N}$, and $\mathbf{D} \in \mathbb{R}^{D\times D}$ are learnable parameters.

The continuous parameters $\mathbf{A}$, $\mathbf{B}$ are converted to discretise forms $\overline{\mathbf{A}}$, $\overline{\mathbf{B}}$ using step size Δ. For digital audio processing, we use zero-order hold to discretise the continuous state space model, which is defined as follows:

$$\overline{\mathbf{A}} = \exp(\Delta\mathbf{A}), \tag{11}$$
$$\overline{\mathbf{B}} = (\Delta\mathbf{A})^{-1}(\exp(\Delta\mathbf{A}) - \mathbf{I}) \cdot \Delta\mathbf{B} \tag{12}$$

As shown in Fig. 1, for block l, the computation follows by

$$r_{l+1} = r_l + \text{DropPath}(h_l), \tag{13}$$
$$h_{l+1} = \text{MambaSSM}(\text{Norm}(r_{l+1})) \tag{14}$$

Here, r_l denotes the residual connection in the layer l, preserving fine-grained temporal information; h_l denotes the hidden state of the layer l, representing the output features of each MambaSSM layer. MambaSSM enables models to adaptively decide which information to retain or forget at each time step, thereby reducing unnecessary parameters. Then we extract tempo-relevant features every three Mamba blocks:

$$f_{tempo} = \frac{1}{|L|}\sum_{l\in L} GlobalAvgPool(h_l) \tag{15}$$

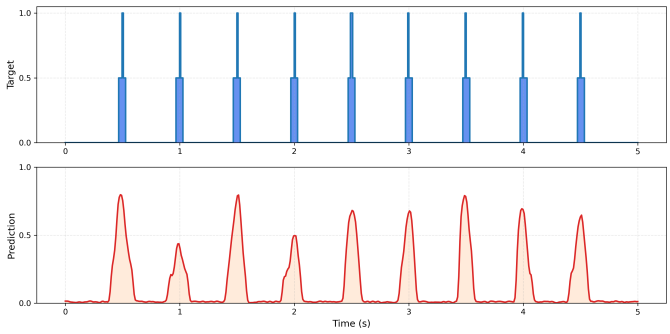

Fig. 2. Beat target and prediction

where $L = \{3, 6, 9\}$ represents layers sampled at three intervals. For each audio frame, we predict beat probability using frame-wise classification in the final hidden states. Formally, the beat probability in the time frame t is defined as:

$$p_{beat}(t) = \text{Softmax}(W_{beat} LayerNorm(h_9(t)) + b_{beat}) \quad (16)$$

where $W_{beat} \in \mathbb{R}^{2 \times d_{model}}$ is the learnable classification weight matrix, $b_{beat} \in \mathbb{R}^2$ is the classification bias vector.

This joint optimization framework enables the model to leverage complementary information from both local rhythmic patterns and global tempo structure during training.

IV. Experiments

A. Datasets and Metrics

We utilize the following six publicly available datasets: Ballroom [30] [31], Hainsworth [32], Carnetic [33], Harmonix [34], SMC [35], and GTZAN [36]. GTZAN dataset is used for testing only, serveing as an additional benchmark for evaluating the generalization capability of the model. Carnetic dataset is employed solely for training to enhance data diversity. As comparable benchmarks neither provide their results nor are open-source, it has been excluded from Table I to ensure fair comparison. The remaining datasets are used for both training and testing purposes in an 8-fold cross-validation manner.

To evaluate the performance of beat tracking, we adopt the following evaluation metrics: F-measure, CMLt and AMLt [37]. F-Measure is our most important evaluation metric and we report it with a tolerance window of ±70 ms. CMLt evaluates the continuous accuracy of beat tracking (using the maximum length of correctly predicted beats). AMLt allows offbeat variations of an annotated beat sequence to be matched with detected beats. CMLt and AMLt measure how well the predicted beat sequences match the ground-truth annotations, indicating the model's capability to maintain consistent rhythmic pattern predictions over time.

B. Experimental Setup

Beat annotations are each represented as a 1D binary sequence that indicates beat (1) and non-beat (0) states at each frame. The joint tracking incorporating tempo branch enhances the performance on beat tracking, and serves as a regularization term that helps improving convergence. We represent annotated beat training data as impulse sequences with the same temporal resolutions as the input features. Accounting for slight deviations in beat position annotation and to address imbalanced number between beat frames and non-beat frames. As shown in Fig. 2, we treat neighboring frames of beat positions as positive samples, but apply weighting adjustment using a scaling factor of 0.5.

Since most audio clips are long, we randomly sample 30-second segments for training. In our experiments, we use BCEWithLogitsLoss as the loss function, using RAdam [38] and LookAhead [39] optimizer with an initial learning rate of 1E-3. Our model has 4.3M trainable parameters and is trained with an NVIDIA GeForce RTX-3090-24GB GPU. For DBN parameters, we optimize `observation_lambda = 8` and `transition_lambda = 2` which achieve the best performance. We train all models for 45 epochs to achieve sufficient convergence and choose the checkpoint with the lowest validation.

TABLE I. Testing results of BeatMamba under 8-fold cross-validation. We use **boldface** to compare with state-of-the-art models and underscore to denote our model.

Dataset	Model	F1	CMLt	AMLt
Ballroom	Ours	96.01	91.61	91.98
	Zhao [25]	**96.80**	**95.40**	96.60
	Hung [13]	96.20	93.90	**96.70**
	Böck [40]	96.20	94.70	96.10
Hainsworth	Ours	**95.19**	**90.71**	93.03
	Zhao [25]	90.40	85.10	**93.70**
	Hung [13]	87.70	86.20	91.50
	Böck [40]	90.20	84.80	93.00
Harmonix	Ours	**97.28**	**94.60**	**96.11**
	Zhao [25]	95.40	90.50	95.70
	Hung [13]	95.30	93.90	95.90
SMC	Ours	**60.56**	44.45	51.21
	Zhao [25]	59.60	45.60	63.50
	Hung [13]	60.50	**51.40**	**66.30**
	Böck [40]	54.40	44.30	63.50
GTZAN	Ours	88.34	79.76	**93.33**
	Zhao [25]	88.50	80.00	92.20
	Hung [13]	**88.70**	81.20	92.00
	Böck [40]	88.50	**81.30**	93.10

C. Results and Analysis

Furthermore, we compare our model with previous works that achieved state-of-the-art performance, including both transformer and neural network models. Table I presents comprehensive evaluation results across five benchmark datasets.

It is obvious that our method outperforms previous approaches in most datasets. Particularly on the Hainsworth dataset, our model demonstrates a 5% improvement in F-measure over the Beat Transformer [25] and 7.5% over SpecTNT [13]. On the Harmonix dataset, BeatMamba achieves state-of-the-art results across all metrics. The superior performance on this modern music dataset indicates that our

dual-channel feature extraction strategy effectively isolates rhythmic information even in heavily produced commercial recordings. On other datasets, although our results do not completely surpass those of other methods, in terms of the total number of parameters and lightweight performance, our model remains nearly optimal. Overall, these results show that our BeatMamba is highly competitive compared to previous SOTA models.

Table II presents efficiency comparisons, revealing significant advantages in resource consumption. With only 4.4M trainable parameters, BeatMamba achieves a 53% reduction compared to Beat Transformer (9.3M), while maintaining better performance across most benchmarks. Our GPU Memory Usage is 5096 MB, representing a 71% reduction compared to Beat Transformer (17,754 MB). This memory efficiency enables deployment on mid-range GPUs and makes real-time processing feasible for low-latency applications. Since SpecTNT–TCN is not open-source, our processing steps may differ from the original implementation, so we do not present its GPU memory usage. Compared to SpecTNT-TCN, our model parameters are reduced by only 6%, yet demonstrating substantial performance advantages: 8% improvement on Hainsworth and 2% improvement on Harmonix. Overall, we have significantly reduced memory usage and parameters while maintaining better performance, creating viable deployment opportunities for real-time or resource-constrained scenarios.

TABLE II. Different models efficiency comparison. "Params" denotes the number of trainable parameters.

Model	Params (M)	GPU Memory (MB)
BeatMamba	**4.4**	**5096**
Beat Transformer [25]	9.3	17754
SpecTNT–TCN [13]	4.7	—

TABLE III. Ablation study of Drum-Separated mechanism and Mamba Block counts on F-measure.

Model	F1-measure
BeatMamba w/o Drum	82.19
BeatMamba-6	85.12
BeatMamba-9	**88.34**
BeatMamba-12	86.94

D. Ablation Study

To evaluate the contribution of each component in BeatMamba, we conduct extensive ablation experiments on the GTZAN dataset using F-Measure as the primary metric. Our ablation study investigates two key aspects: the drum-separated mechanism and the number of Mamba blocks.

In Table III, we compare the following two groups: (a) BeatMamba w/o Drum; (b) BeatMamba-Nr, where Nr denotes the number of Mamba blocks. In the first group, each model shares the same architecture with nine Mamba blocks, differing only in whether a drum track is utilized. As shown in Table III, we can see that F-measure when using drum is nearly 6% higher than without drum. Experimental results demonstrate that the drum-separated mechanism effectively enhances beat tracking accuracy. The improvement arises from the drum channel, whose percussive signals supply clear rhythmic structure and enhance features in the mixed audio. By isolating the drum component, the model can focus more effectively on rhythmically salient elements, reducing interference from harmonic and melodic content. In the second group, we tested the impact of the number of Mamba blocks on model's performance. We evaluated BeatMamba-6, BeatMamba-9, and BeatMamba-12. The results indicate that the optimal model is 9 blocks. We observed validation loss across different layer counts and the model converged best at 9 layers. The 6-block model shows decent performance, but fails to fully leverage the complementary information from dual-channel. The limited depth restricts the model's capacity to capture long-range temporal dependencies. The degradation in the 12-block model can be attributed to gradient decay and optimization challenges associated with excessive depth. We measured gradient norms across layers and found that the gradient ratio (first layer/last layer) drops significantly from 0.58 in the 9-block configuration to 0.23 in the 12-block model. This reduction indicates severe gradient decay in deeper architectures, making it difficult for learning signals to be effectively propagated to earlier layers.

V. Conclusions

In this paper, we propose BeatMamba, an efficient beat tracking method that combines state space models with a dual-channel input mechanism. Our approach leverages the linear complexity of Mamba blocks to significantly reduce computational costs while maintaining competitive performance compared to transformer-based models. The dual-channel strategy, incorporating both original mix and drum-separated tracks, effectively simulates human beat perception by providing complementary rhythmic information for enhanced beat detection. Experiments demonstrate our superior efficiency and competitive performance compared to existing models. These promising results suggest that state space models have significant potential for advancing beat tracking analysis. Future research will focus on extending BeatMamba to real-time streaming scenarios, incorporating multi-stem separation strategies and music foundation models.

Acknowledgment

We extend our appreciation to all reviewers of WCCI 2026 for their thorough reading of our manuscript and for their valuable comments.

References

[1] G. Alagöz, E. Eising, Y. Mekki *et al.*, "The shared genetic architecture and evolution of human language and musical rhythm," *Nature Human Behaviour*, vol. 9, pp. 376–390, 2025.

[2] K. Ochiai, H. Kameoka, and S. Sagayama, "Explicit beat structure modeling for non-negative matrix factorization-based multipitch analysis," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2012, pp. 133–136.

[3] R. Nishikimi, E. Nakamura, M. Goto, K. Itoyama, and K. Yoshii, "Bayesian singing transcription based on a hierarchical generative model of keys, musical notes, and f0 trajectories," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 1678–1691, 2020.

[4] G. Shibata, R. Nishikimi, and K. Yoshii, "Music structure analysis based on an lstm-hsmm hybrid model," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2020, pp. 23–29.

[5] T. Kim and J. Nam, "All-in-one metrical and functional structure analysis with neighborhood attentions on demixed audio," in *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. IEEE, 2023, pp. 1–5.

[6] A. H. D. Crooke and K. S. McFerran, "Improvising using beat making technologies in music therapy with young people," *Music Therapy Perspectives*, vol. 37, no. 1, pp. 55–64, 2019.

[7] M. Heydari, J.-C. Wang, and Z. Duan, "Singnet: A real-time singing voice beat and downbeat tracking system," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.

[8] S. Durand, J.-P. Bello, B.David, and G.Richard, "Robust downbeat tracking using an ensemble of convolutional networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 1, pp. 76–89, 2016.

[9] E. P. MatthewDavies and S. Böck, "Temporal convolutional networks for musical audio beat tracking," in *27th European Signal Processing Conference, EUSIPCO 2019, A Coruña, Spain, September 2-6, 2019*. IEEE, 2019, pp. 1–5.

[10] F. Krebs, S. Böck, M. Dorfer, and G. Widmer, "Downbeat tracking using beat synchronous features with recurrent neural networks," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2016, pp. 129–135.

[11] 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)*, 2016, pp. 255–261.

[12] M. Heydari, F. Cwitkowitz, and Z. Duan, "Beatnet: Crnn and particle filtering for online joint beat downbeat and meter tracking," *arXiv preprint arXiv:2108.03576*, 2021.

[13] Y.-N. Hung, J.-C. 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)*, 2022, pp. 401–405.

[14] T. Cheng and M. Goto, "Transformer-based beat tracking with low-resolution encoder and high-resolution decoder," in *Proceedings of the 24th International Society for Music Information Retrieval Conference, ISMIR 2023, Milan, Italy, November 5 -9, 2023*, 2023, pp. 466–473.

[15] C.-C. Chang and L. Su, "Beast: Online joint beat and downbeat tracking based on streaming transformer," in *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2024, pp. 396–400.

[16] L. Zhu, B. Liao, Q. Zhang, X. Wang, W. Liu, and X. Wang, "Vision mamba: Efficient visual representation learning with bidirectional state space model," in *Forty-first International Conference on Machine Learning, ICML 2024,Vienna, Austria, July 21-27, 2024*, 2024.

[17] A. Gu, I. Johnson, K. Goel, K. Saab, T. Dao, A. Rudra, and C. Ré, "Combining recurrent, convolutional, and continuous-time models with linear state space layers," in *NeurIPS*, 2021, pp. 572–585.

[18] 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)*, 2010, pp. 649–654.

[19] S. Vardhan, P. R. Acharya, S. S. Rao, O. R. Jasthi, and Natarajan, "An ensemble approach to music source separation: A comparative analysis of conventional and hierarchical stem separation," in *EvoMUSART*, 2025.

[20] A. Bellia, "Introduction. percussion instruments in the ancient world: Towards an archaeology of musical performance," *Pallas. Revue d'études antiques*, no. 115, pp. 09–23, 2021.

[21] M. Goto, "An audio-based real-time beat tracking system for music with or without drum sounds," *Journal of New Music Research*, vol. 30, no. 2, pp. 159–171, 2001.

[22] S. Dixon, "Automatic extraction of tempo and beat from expressive performances," *Journal of New Music Research*, vol. 30, no. 1, pp. 39–58, 2001.

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

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

[25] J. Zhao, G. Xia, and Y. Wang, "Beat transformer: Demixed beat and downbeat tracking with dilated self-attention," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2022.

[26] A. Gagneré, S. Essid, and G. Peeters, "A contrastive self-supervised learning scheme for beat tracking amenable to few-shot learning," in *Proceedings of the 25th International Society for Music Information Retrieval Conference, ISMIR 2024, San Francisco, California, USA and Online, November 10-14, 2024*, 2024, pp. 198–206.

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

[28] 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 2016 ACM Conference on Multimedia Conference, MM 2016, Amsterdam, The Netherlands, October 15-19, 2016*, 2016, pp. 1174–1178.

[29] S. Rouard, F. Massa, and A. Défossez, "Hybrid transformers for music source separation," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023, pp. 1–5.

[30] F. Gouyon, A. Klapuri, S. Dixon, M. Alonso, G. Tzanetakis, C. Uhle, and P. Cano, "An experimental comparison of audio tempo induction algorithms," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 5, pp. 1832–1844, 2006.

[31] F. Krebs, S. Böck, and G. Widmer, "Rhythmic pattern modeling for beat and downbeat tracking in musical audio." in *Ismir*, 2013, pp. 227–232.

[32] S.W. Hainsworth and M.D. Macleod, "Particle filtering applied to musical tempo tracking," *EURASIP Journal on Advances in Signal Processing*, vol. 2004, no. 15, pp. 1–11, 2004.

[33] A. Srinivasamurthy and X. Serra, "A supervised approach to hierarchical metrical cycle tracking from audio music recordings," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 5217–5221.

[34] O. Nieto, M. McCallum, M.E.P. Davies, A. Robertson, A. M. Stark, and E. Egozy, "The harmonix set: Beats, downbeats, and functional segment annotations of western popular music," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2019, pp. 565–572.

[35] A. Holzapfel, M.E.P. Davies, J.R. Zapata, J. L. Oliveira, and F. Gouyon, "Selective sampling for beat tracking evaluation," *IEEE Transactions on Speech and Audio Processing*, vol. 20, no. 9, pp. 2539–2548, 2012.

[36] U. Marchand, Q. Fresnel, and G. Peeters, "Gtzan-rhythm: Extending the gtzan test-set with beat, downbeat and swing annotations," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2015.

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

[38] L. Liu, H. Jiang, P. He, W. Chen, X. Liu, J. Gao, and J. Han, "On the variance of the adaptive learning rate and beyond," in *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*, 2020.

[39] M.R. Zhang, J. Lucas, J. Ba, and G.E. Hinton, "Lookahead optimizer: k steps forward, 1 step back," in *Advances in Neural Information Processing Systems*, vol. 32, 2019, pp. 9593–9604.

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