BEATNET: CRNN AND PARTICLE FILTERING FOR ONLINE JOINT BEAT DOWNBEAT AND METER TRACKING

Mojtaba Heydari **Frank Cwitkowitz** **Zhiyao Duan**
Department of Electrical and Computer Engineering
University of Rochester, 500 Wilson Blvd, Rochester, NY 14627, USA
{mheydari,fcwitkow}@ur.rochester.edu zhiyao.duan@rochester.edu

ABSTRACT

The online estimation of rhythmic information, such as beat positions, downbeat positions, and meter, is critical for many real-time music applications. Musical rhythm comprises complex hierarchical relationships across time, rendering its analysis intrinsically challenging and at times subjective. Furthermore, systems which attempt to estimate rhythmic information in real-time must be causal and must produce estimates quickly and efficiently. In this work, we introduce an online system for joint beat, downbeat, and meter tracking, which utilizes causal convolutional and recurrent layers, followed by a pair of sequential Monte Carlo particle filters applied during inference. The proposed system does not need to be primed with a time signature in order to perform downbeat tracking, and is instead able to estimate meter and adjust the predictions over time. Additionally, we propose an information gate strategy to significantly decrease the computational cost of particle filtering during the inference step, making the system much faster than previous sampling-based methods. Experiments on the GTZAN dataset, which is unseen during training, show that the system outperforms various online beat and downbeat tracking systems and achieves comparable performance to a baseline offline joint method.

1. INTRODUCTION

Rhythm plays an essential role in nearly all musical endeavors, including listening to, playing, learning, or composing music. This is why the estimation of rhythmic information, such as beat positions, downbeat positions and meter has always been an important subject of study in the field of Music Information Retrieval (MIR). Depending on the requirements and constraints imposed by the application at hand, these estimation tasks can either be performed in an offline or online fashion. Offline approaches are typically non-causal, meaning that they make predictions for a given time using data or features associated with a future time. These approaches are suitable for applications such as music transcription, music search and indexing, and musicological analysis. Online approaches are causal, meaning that they operate using only past and present features. These are typically desirable for human-computer interaction (HCI) systems, which must make immediate predictions, like real-time music accompaniment systems.

Many offline methods have been proposed for beat tracking [1–3]. Most of them are unsupervised and attempt to utilize low-level features like onset strengths with some inference model to estimate beat positions within a music piece. However, with the growing success of deep learning, supervised beat tracking methods have become more prominent. Böck et al. [4] employed Recurrent Neural Networks (RNNs) to estimate beat positions; Various other neural network structures have also been proposed for onset detection and beat tracking [5, 6].

Some methods have also been proposed for online beat tracking. However, many of them, e.g., [4, 7–10], feed a sliding window of data into an offline model to estimate beat positions within upcoming frames. The sliding window strategy has several major drawbacks, including the discontinuity of beat predictions and the need for priming for predictions in the first window, which causes a delay [11]. Some other approaches involve inferring beat positions in real time using multi agent models [11–14], which initialize a set of agents with various hypotheses that try to validate their respective hypotheses based on observations across time.

The task of downbeat tracking is often considered to be more difficult than beat tracking. This is because a deeper understanding of rhythmic structure in music is required to be able to differentiate between beats and downbeats. Making matters worse, at the signal level, these two events have very similar characteristics. For instance, downbeats are not necessarily associated with stronger signal energy, nor do they necessarily feature a distinct percussive profile. Moreover, both beats and downbeats are likely to be the intersection of melodic and harmonic changes. These factors can make it challenging, and in some cases subjective, to distinguish between the two rhythmic events. For instance, for a 4/4 music piece with kick drum events on the first and third beats, it is hard to distinguish downbeats and determine whether the time signature is 4/4 or 2/4.

There has been some previous work on offline downbeat tracking, both as an isolated task and within a joint beat and downbeat tracking framework. Durand et al. [15–

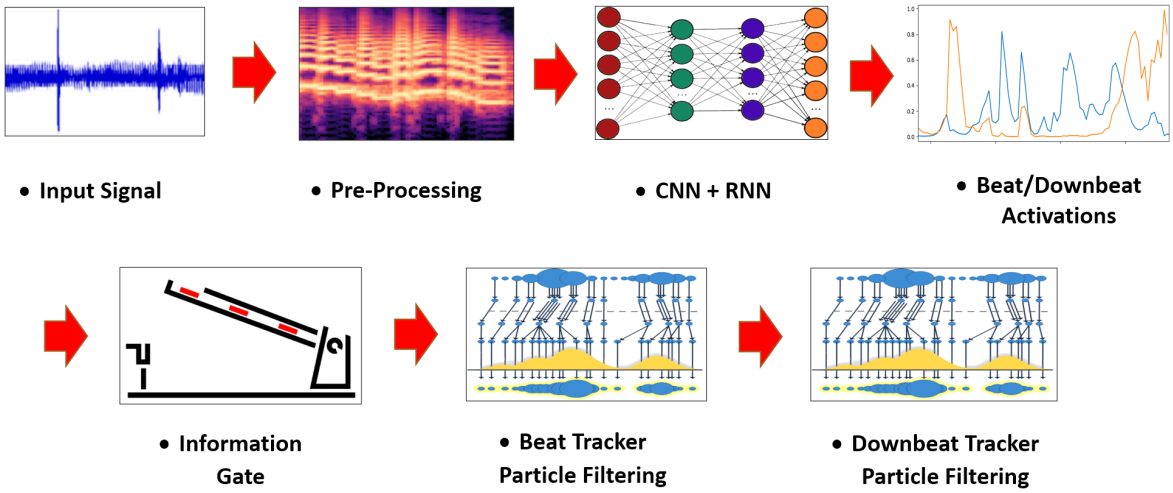

Figure 1. Overview of the joint beat, downbeat, and meter tracking procedure using the proposed BeatNet model.

17] used some combinations of features and CNN structures to obtain downbeats. Giorgi et al. [18] proposed tempo-invariant convolutional filters for downbeat tracking. Peeters and Papadopoulos [19] performed joint beat and downbeat tracking by decoding hidden states using the Viterbi algorithm. Böck et al. [20] and Krebs et al. [21] employed an RNN structure for joint beat and downbeat tracking and only downbeat tracking using beat synchronous features, respectively. Furthermore, some recent works investigate Convolutional Recurrent Neural Network (CRNN) structures for beat and downbeat tracking. Fuentes et al. [22] showed that CRNN structures outperform RNNs in downbeat tracking when taking the input observations over a tatum grid. Cheng et al. [23] found that CRNN structures with larger receptive fields outperform other downbeat tracking models. Böck and Davies. [24] used a CNN and Temporal Convolutional Network (TCN) structure to improve the performance of their offline beat and downbeat tracking model, and also performed data augmentation to expose the neural network to more tempi.

The task of online downbeat tracking has received considerably less attention. Goto and Muranoka [13] introduced an unsupervised model which leverages a measure inference stage for detecting chord changes. In [25], the same beat tracking neural network with forward algorithm from [4, 20] is paired with [21] to estimate downbeats and other rhythmic patterns by extracting percussive and harmonic beat-synchronous features. It is important to note that this method must be primed with a known time signature and all possible rhythmic pattern choices. Liang [26] proposed an online downbeat tracking method which feeds a sliding window of data to an offline model [17]. This method is vulnerable to the sliding window strategy drawbacks described above.

Particle filtering is advantageous for two main reasons when it comes to online processing. The first reason is that it does not require future data. Popular maximum a posteriori (MAP) algorithms like the Viterbi algorithm and maximizer of the posterior marginals (MPM) smoothing algorithms, e.g. forward-backward, are not applicable to online processing. The second reason is that, among the filtering methods which are causal, particle filtering is a general (non-parametric) approach which can be utilized to decode any unknown distribution. However, most music rhythmic analysis approaches that utilize particle filtering, e.g., [27–30], are classical and do not incorporate neural networks. Alternatively, in our previous work [31], we utilized a particle filtering inference model to infer beat positions using the activations produced by an RNN in an online fashion, but that approach does not attempt to estimate downbeats nor meter.

In this paper, we propose BeatNet, a novel online system for joint beat, downbeat, and meter tracking. The system produces beat and downbeat activations using a CNN and RNN combination, and performs inference using two particle filtering stages. The beat tracking stage outperforms state-of-the-art online beat tracking methods. The other stage simultaneously infers downbeats and time signature and achieves comparable results to state-of-the-art offline downbeat tracking models that require the time signature as input. In contrast, BeatNet actively monitors tempo and time signature changes over time. Finally, we introduce an information gate mechanism in the inference module to speed up the inference significantly, making our method suitable for many real-time applications.

2. METHOD

In this section, we describe BeatNet, our online system for joint beat, downbeat, and meter tracking, illustrated in Figure 1. BeatNet consists of a causal neural network stage for producing activations and a particle filtering stage for inference. The neural network comprises convolutional, recurrent and fully connected layers as described in section 2.2 which compute beat and downbeat activations for each frame of audio. The activations are fed to a two-stage particle filtering module to infer beat and downbeat positions and to estimate meter. The code for the BeatNet model is open-source [1], along with video demos and further docu-

[1] https://github.com/mjhydri/BeatNet

mentation.

2.1 Feature Representation

The input of the network module is a sequence of filterbank magnitude responses, each of which corresponds to one audio frame. Specifically, short-time Fourier transform (STFT) with a Hann window of the length of 93 ms and hop size of 46 ms is applied to the audio signal to compute the log-amplitude magnitude spectrogram. Then a logarithmically spaced filterbank ranging from 30 Hz to 17 kHz with 24 bands per octave is applied to yield a 136-d filterbank response. The first-order temporal difference of this response is also calculated and concatenated, resulting in a 272-d filterbank response vector for each frame.

We also experimented with alternative feature representations, including the 329-d hand-crafted feature set from [15], which comprises chroma features, onset strengths, low-frequency spectral features, and melodic constant-Q spectral features. The motivation for this feature set is to aggregate the harmonic, percussive, bass, and melodic content of the music. However the 272-d filterbank response feature set described above achieved notably better performance than these hand-crafted features, and was thus chosen for subsequent experiments.

2.2 Network Architecture

Following the common design of other similar works, we employ a convolutional-recurrent neural network (CRNN) architecture, illustrated in Figure 2, to process the input features in order to obtain beat and downbeat activations. Ideally, the convolution models relationships along the frequency axis, and the unidirectional recurrence models long-term relationships across time in a causal fashion.

The input features are fed into a 1D convolutional layer with 2 filters of kernel size 10, followed by ReLU activation. The two filter responses are max pooled with kernel size 2 along frequency and then concatenated into a single feature embedding for each frame. Then, a fully-connected layer with 150 neurons reduces the dimensionality of the embedding, and feeds it through two subsequent unidirectional Long Short-Term Memory (LSTM) layers, each with a hidden size of 150. The embedding is then fed through a final fully-connected layer and a softmax operation to obtain three activations which represent beat, downbeat, and non-beat, respectively. Note that due to the softmax function, the final activations for each class always sum to one.

2.3 Particle Filtering Inference

In this section, we discuss the two-stage online Monte Carlo particle filtering inference module, which generates the beat and downbeat predictions. Sequential Monte Carlo particle filtering is a sampling-based model which iteratively estimates any unknown distribution $p(x)$ by gathering a large number of independent samples from an arbitrary proposal distribution. The unknown distribution of

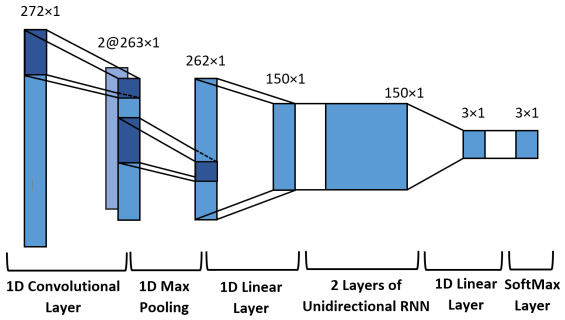

Figure 2. Proposed CRNN architecture for processing input features and computing beat and downbeat activations.

interest in our case, up to the K-th frame, is the following posterior $p(x_{1:K}|y_{1:K})$ of underlying beat or downbeat positions $x_{1:K}$ conditioned on beat observations $y_{1:K}$. It can be inferred according to the key equations below. For more detailed information, please refer to our previous work [31].

$$p(x) = \lim_{N\to\infty} \sum_{i=1}^{N} \frac{\omega^{(i)}}{\sum_{i=1}^{N} \omega^{(i)}} \delta\left(x - x^{(i)}\right), \quad (1)$$

$$p\left(x_{1:K}^{(i)}|y_{1:K}\right) \propto \prod_{k=1}^{K} p\left(y_k|x_k^{(i)}\right) p\left(x_k^{(i)}|x_{k-1}^{(i)}\right), \quad (2)$$

$$\omega_k^{(i)} = p\left(y_k|x_k^{(i)}\right) \omega_{k-1}^{(i)}, \quad (3)$$

where $\omega^{(i)}$ is the importance weight of particle i, and $\delta(\cdot)$ is the Dirac function. Eq. (1) describes the estimation of $p(x)$ using a large number of particles $(N \to \infty)$ and their importance weights. Eq. (2) is a dynamic model which updates the posterior of each frame k using the transition (motion) and observation (correction) probabilities. Eq. (3) describes a recursive process to update the importance weights using the current observation and the importance weights of the previous step.

2.3.1 State spaces, transition and observation models

We use a cascade of two sequential Monte Carlo particle filters, one for beat tracking, and the other for downbeat and meter tracking. The state space and transition model of the beat estimator are similar to [32]. The beat state space is a type of 2D bar pointer model and its transition for the phase (horizontal) and the tempo (vertical) of the frame are described in Eqs. (4) and (5), respectively. The phase of frame k within a beat interval and the tempo at frame k are respectively denoted by $\phi_{b,k}$ and $\dot{\phi}_{b,k}$. A constant λ_b influences the intensity of potential jumps across the tempo axis.

We propose a new beat observation model in Eq. (6), where $x_{b,k}$ and $y_{b,k}$ are the beat state and beat observations at frame k. For non-beat states we allocate a small likelihood as $\gamma = 0.03$ instead of using the non-beat activation output from the neural network. For beat frames, since downbeats can also be considered beats, we assess

the maximum of the beat and downbeat activations. If the maximum exceeds a certain threshold, i.e., $T = 0.4$, then it is set as the likelihood; Otherwise, γ is used. When γ is used, we also bypass the costly re-sampling step in the beat particle filtering module. Therefore, the threshold serves as an *information gate*, through which the computational cost is significantly reduced.

$$\phi_{b,k} = (\phi_{b,k-1} + \dot{\phi}_{b,k-1}) \mod (\phi_b^{max} + 1), \quad (4)$$

$$p(\dot{\phi}_{b,k}|\dot{\phi}_{b,k-1}) = \begin{cases} \exp\left(-\lambda_b \left|\frac{\dot{\phi}_{b,k}}{\dot{\phi}_{b,k-1}}\right|\right) & \text{if } \phi_{b,k} = 0 \\ \mathbb{1}(\dot{\phi}_{b,k} = \dot{\phi}_{b,k-1}) & \text{if } \phi_{b,k} > 0 \end{cases}, \quad (5)$$

$$p(y_{b,k}|x_{b,k}) = \begin{cases} \max(b_k, d_k) & \text{if } \phi_{b,k} = 0 \text{ and} \\ & \max(b_k, d_k) \geq T \\ \gamma & \text{otherwise} \end{cases}, \quad (6)$$

The second particle filter detects downbeats and the time signature jointly. The state space is similar to that of beat tracking. However, here we introduce $\dot{\phi}_{d,k}$ corresponding to the meter, i.e., $\dot{\phi}_{d,k} \in 2, 3, ..., \dot{\phi}_d^{max}$, and $\phi_{d,k}$ to describe the phase of the beat within the bar interval, i.e. $\phi_{d,k} \in 0, 1, 2, ..., \phi_d^{max}$. Eqs. (7) and (8) describe the phase and meter transition models. We only let meter change at the states belonging to the downbeat area i.e. $\phi_{d,k} = 0$, and λ_d is a constant parameter that decides what percent of the particles jump to other meters at the downbeat states. Also, in Eq. (9) we define the observation model used in the downbeat particle filter. The first states within the bar (downbeat area) take the downbeat activation and the rest of them (beat states) take the beat activation. Note that as the second particle filter operates less often, i.e., only when a beat is detected, no information gate is needed here.

$$\phi_{d,k} = (\phi_{d,k-1} + \dot{\phi}_{d,k-1}) \mod (\phi_d^{max} + 1), \quad (7)$$

$$p(\dot{\phi}_{d,k}|\dot{\phi}_{d,k-1}) = \begin{cases} \lambda_d & \text{if } \phi_{d,k} = 0 \text{ and} \\ & \dot{\phi}_{d,k} \neq \dot{\phi}_{d,k-1} \\ 1 - \lambda_d & \text{if } \phi_{d,k} = 0 \text{ and} \\ & \dot{\phi}_{d,k} = \dot{\phi}_{d,k-1} \\ \mathbb{1}(\dot{\phi}_{d,k} = \dot{\phi}_{d,k-1}) & \text{if } \phi_{d,k} > 0 \end{cases}, \quad (8)$$

$$p(y_{d,k}|x_{d,k}) = \begin{cases} d_k & \text{if } \phi_{d,k} = 0 \\ b_k & \text{if } \phi_{d,k} > 0 \end{cases}, \quad (9)$$

2.3.2 Inference process

Algorithm 1 describes the inference process in detail. Particles are initialized randomly for both inference modules by sampling from a uniform distribution within their state space. By proceeding to a new frame, particles within the beat state space are transferred to the new positions by sampling from the transition model, and new importance weights are then calculated and normalized. If the activations of the frame satisfy the information gate condition, the re-sampling process is invoked for all particles; Otherwise, the re-sampling step is skipped as it is likely a non-beat frame. Afterwards, if the median of the particles is within the tolerance window T_w of a beat area and the time of the current frame is longer enough than the last detected beat considering the estimated tempo, the frame is classified as a beat frame. A similar process follows for the downbeat and meter inference module.

Algorithm 1 Joint Inference Procedure

beats, downbeats, meters = [], [], []
Sample $(x_{b,0}^{(i)}) \sim \mathcal{U}(S_b)$, $(x_{d,0}^{(j)}) \sim \mathcal{U}(S_d)$
Set $w_{b,0}^{(i)} = \frac{1}{N_b}$, $w_{d,0}^{(j)} = \frac{1}{N_d}$
for $k = 1$ to K **do**
  Sample $(x_{b,k}^{(i)}) \sim p(\phi_{b,k}^{(i)}|\phi_{b,k-1}^{(i)}), p(\dot{\phi}_{b,k}^{(i)}|\dot{\phi}_{b,k-1}^{(i)})$
  $\tilde{\omega}_{b,k}^{(i)} = \omega_{b,k-1}^{(i)} \times p(y_{b,k}|x_{b,k}^{(i)}) \quad \forall i \in N_b$
  $\omega_{b,k}^{(i)} = \frac{\tilde{\omega}_{b,k}^{(i)}}{\sum \tilde{\omega}_{b,k}^{(i)}} \quad \forall i \in N_b$
  if $\max(b_k, d_k) \geq T$ **then**
    Resample $x_{b,k}^{(i)}$ according to $\omega_{b,k}^{(i)}$
  end if
  if median$(\phi_{b,k}^{(i)}) < T_w$ **and** $(k\Delta - beats[-1]) > 0.4$ median$(\dot{\phi}_{b,k}^{(i)})$ **then**
    Append (beats, $k\Delta$)
    Sample $(x_{d,k}^{(j)}) \sim p(\phi_{d,k}^{(j)}|\phi_{d,k-1}^{(j)}), p(\dot{\phi}_{d,k}^{(j)}|\dot{\phi}_{d,k-1}^{(j)})$
    $\tilde{\omega}_{d,k}^{(j)} = \omega_{d,k-1}^{(j)} \times p(y_{d,k}|x_{d,k}^{(j)}) \quad \forall j \in N_d$
    $\omega_{d,k}^{(j)} = \frac{\tilde{\omega}_{d,k}^{(j)}}{\sum \tilde{\omega}_{d,k}^{(j)}} \quad \forall j \in N_d$
    Resample $x_{d,k}^{(j)}$ according to $\omega_{d,k}^{(j)}$
    if mode$(\phi_{d,k}^{(j)}) == 0$ **then**
      append (downbeats, $k\Delta$)
      append (meters, mode$(\dot{\phi}_{d,k}^{(j)})$)
    end if
  end if
end for

A visualization of the inference process is presented in Figure 3. Each pair of plots demonstrates one step of the inference procedure, where the top and the bottom plots show the beat and downbeat tracking process, respectively. In the first pair of plots, the beat particles are initialized randomly. In the second pair, the first beat is detected and the downbeat state particles are simultaneously initialized randomly. In the third pair, beat tracking particles have converged, but the downbeat particles have not yet converged. Here the downbeat clutter is located in the lowest row of the downbeat state space, which represents a six-beat time signature. The next few plot pairs illustrate convergence of both the beat and downbeat particles, producing an estimate of the tempo and beat phase (top plots), and the meter and bar phase (bottom plots).

3. EXPERIMENTS

3.1 Methodology

In order to analyze the performance of BeatNet, we compare it to several publicly available online beat tracking methods, We additionally provide the online downbeat tracking performance of BeatNet for each of the experiments. Following standard evaluation practices, in

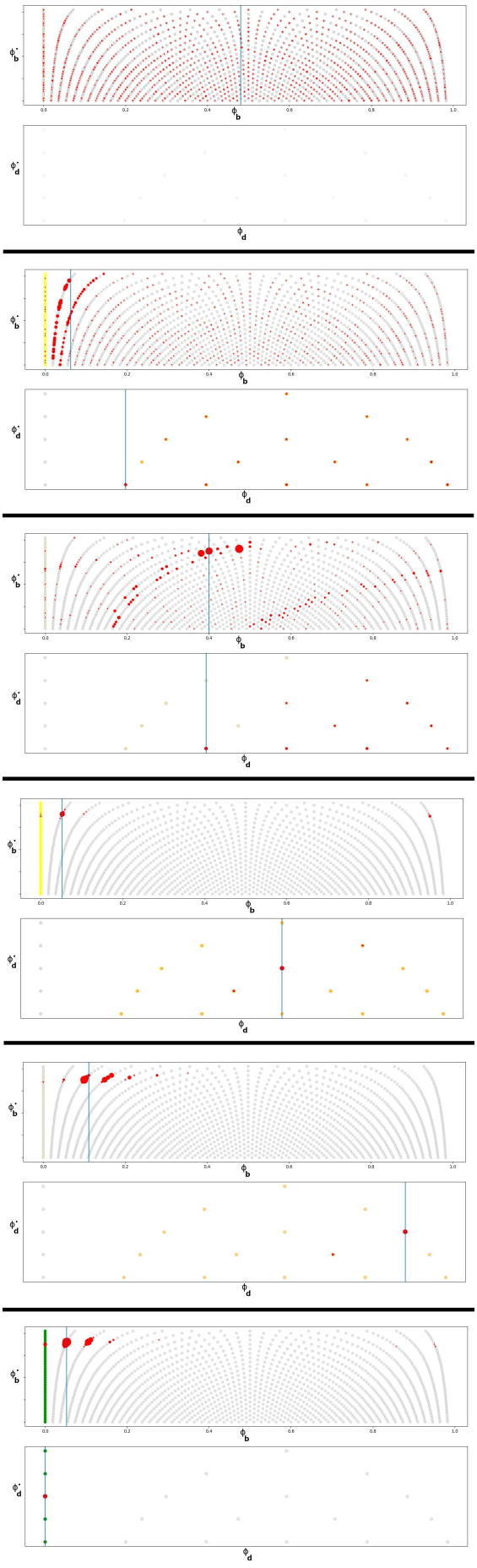

Figure 3. Inference example, detailed in Section 2.3.2.

Dataset	*# Files*	*Total Length*
Ballroom [33, 34]	685	5 h 57 m
Beatles [2]	180	8 h 9 m
Carnatic [35]	176	16 h 38 m
GTZAN [36, 37]	999	8 h 20 m
Rock Corpus [38]	200	12 h 53 m

Table 1. Datasets used for training and testing.

this work, F-measure with a tolerance window of $T_w = \pm 70\ ms$ is used as the evaluation metric for all experiments.

We utilize all five datasets [2, 33–38] described in Table 1 for training, validation, and testing, with different splits and arrangements for various experiments. In the first comparison, we evaluate BeatNet on the GTZAN dataset, which covers 10 different music genres and was unseen from training of all comparison methods. In order to demonstrate the generalization ability of our approach, we also experiment with two other comparison schema where we respectively set aside the Ballroom and Rock datasets during training and use them entirely for evaluation. Note that all of the supervised comparison methods included the Ballroom and Rock datasets in their training set, so we only compare BeatNet with unsupervised methods in these cases.

3.2 Training Details

For training the beat and downbeat activation neural network described in Section 2.2, all weights and biases are initialized randomly, and the network is trained using Adam optimizer with a learning rate of 5×10^{-4} and a batch size of 200. Since the number of non-beat frames within a music piece is typically much larger than the number of beat and downbeat frames, our objective function is chosen to be weighted cross entropy loss of the beat, downbeat, and non-beat, where the weights are inverse proportional to the frequency of occurrence of each type of frame. Batches comprise 8-second long excerpts randomly sampled from each audio file available in the training set. Given that some datasets (e.g., Beatles) contain full songs and others (e.g., Ballroom) contain short excerpts of songs, we sample from longer audio files more often during the training Batch creation. Training proceeds until the performance on the validation set has not increased over a span of 20 epochs for a given experiment.

3.3 Results and Discussion

The evaluation results of the proposed BeatNet model and comparison methods are presented in Table 2. All online comparison methods only perform beat tracking, and all except IBT [11] and Aubio [9] are supervised methods using deep neural networks. We can see that the online beat tracking portion of BeatNet outperforms all comparison methods. The Böck FF [6, 20] and Don't Look Back (DLB) models [31] achieve the next best performance.

Method	*F-Measure Beats*	*F-Measure Downbeats*
Comparison of Online Methods		
GTZAN Dataset		
Aubio [9]	57.09	—
BeatNet	75.44	46.49
Böck ACF [4]	64.63	—
Böck FF [6, 20]	74.18	—
DLB [31]	73.77	—
IBT [11]	68.99	—
Ballroom Dataset		
Aubio [9]	56.73	—
BeatNet	77.41	47.45
IBT [11]	70.79	—
Rock Corpus Dataset		
Aubio [9]	59.83	—
BeatNet	73.13	44.98
IBT [11]	68.55	—
Comparison of Offline Methods		
GTZAN Dataset		
BeatNet + DBN	80.64	54.07
Böck [20]	79.09	51.36

Table 2. Comparison of BeatNet with other beat and downbeat tracking methods on various datasets.

Böck FF uses the forward algorithm to estimate beats in a similar manner to the other online joint model described earlier [25]. Aside from the different neural network structures, the beat tracking inference processes of the DLB model [31] and BeatNet are largely the same. The main difference is that the latter benefits from the information gate, which decreases the computational time drastically.

Additionally, we report the performance comparison with an offline joint beat and downbeat tracking model [20] on the GTZAN dataset. In this case, we replaced the particle filtering modules of BeatNet with the DBN used in [20] to directly compare neural network architectures in BeatNet and [20]. Same to [20], we also provided the time signatures to the DBN. For [20], we utilized the Madmom [39] library, which is the official implementation of the paper. Note that due to the existence of different GTZAN beat annotations, the reported offline results obtained by us differ from those of the original paper [20]. However, since we used the same annotations for all of the experiments, the offline comparison is valid. As the table suggests, with the same DBN estimator, both neural networks yield similar results for beat tracking. However, for downbeat tracking, the BeatNet architecture yields marginally better performance. These results are interesting, since we are comparing a causal network to a non-causal network which leverages bidirectional recurrence. However, our network is larger and contains more parameters.

The comparison between BeatNet (second row) and [20] (last row) is also interesting. BeatNet underperforms [20] by 3.65% on beat tracking and by 4.9% on downbeat tracking. However, it is noted that BeatNet is an online method and it does not require the time signature input, while [20] is offline method and it requires the time signature input.

One limitation of our model is that the performance of the downbeat tracker depends on the beat tracker. This means that if the beat tracker makes incorrect predictions, errors will carry through to the downbeat tracker. This is a common characteristic of cascade systems such as [25]. Another limitation is the high computation cost of sequential Monte Carlo particle filtering methods. This limitation has been partially addressed in our previous work [31] by using efficient models, e.g., [32] in the inference stage. The information gate proposed in this paper further reduces the computational cost.

On a typical windows machine with AMD Ryzen 9 3900X CPU and 3.80 GHz clock, the processing time for the pre-processing stage and passing a frame through the neural network is 0.12 ms and 0.01 ms, respectively. These times are relatively insignificant, as the inference process takes more time. The inference process takes 5.23 and 8.87 seconds using 1000 and 1750 particles, respectively, to process a 30-sec long music excerpt. This is much faster than the previous sampling-based model [31] which took 21.30 seconds using a 1000 particle setup. Larger numbers of particles lead to longer processing times with a roughly linear relationship. Hence, we reported these results using 1500 particles for the beat inference block and 250 for the downbeat inference block (1750 particles in total) to keep the process minimal.

4. CONCLUSION

We proposed BeatNet, a new online system for joint beat, downbeat, and meter tracking. The system incorporates a convolutional-recurrent neural network for generating beat and downbeat activations in each audio frame, and a two-stage particle filtering algorithm to estimate tempo, beats, downbeats, and musical meter. An information gate is added to the beat tracking particle filter to skip many re-sampling steps hence reduces the computational cost significantly. The system is compared to multiple online and offline methods under various experimental conditions, and it achieves superior performance for both online beat and downbeat tracking.

5. ACKNOWLEDGEMENT

This work has been partially supported by the National Science Foundation grants 1846184 and DGE-1922591.

6. REFERENCES

[1] D. Ellis, "Beat tracking with dynamic programming," *Journal of New Music Research*, vol. 36, no. 1, pp. 51–60, 2007.

[2] M. E. P. Davies, N. Degara, and M. D. Plumbley, "Evaluation methods for musical audio beat tracking algorithms," in *Technical Report C4DM-TR-09-06*,

Centre for Digital Music, Queen Mary University of London, 2009.

[3] F. Gouyon, *A computational approach to rhythm description Audio features for the computation of rhythm periodicity functions and their use in tempo induction and music content processing.* PhD thesis, Universitat Pompeu Fabra, 2005.

[4] S.Böck and S. Schedl, "Enhanced beat tracking with context-aware neural networks," in *In Proc. ofthe 14th International Conference on Digital Audio Effects (DAFx-11)*, 2011, pp. 135–140.

[5] M. E. P. Davies and S.Böck, "Temporal convolutional networks for musical audio beat tracking," in *In Proc. of the 27th European Signal Processing Conference (EUSIPCO)*, 2019.

[6] S.Böck, F. Krebs, and G.Widmer, "A multi-model approach to beat tracking considering heterogeneous music styles," in *In Proc. of the 15th Intl. Conf. on Music Information Retrieval (ISMIR)*, 2014, pp. 603–608.

[7] M. E. P. Davies, P. M. Brossier, and M. D. Plumbley, "Beat tracking towards automatic musical accompaniment," *Audio Eng. Soc. Conv. Spring Prepr*, vol. 2, pp. 751–757, 2005.

[8] A. Gkiokas and V. Katsouros, "Convolutional neural networks for real-time beat tracking: A dancing robot application," in *In Proc. of the 18th Intl. Conf. on Music Information Retrieval (ISMIR)*, 2017, pp. 286–293.

[9] P. M. Brossier, "Automatic annotation of musical audio for interactive applications," P. dissertation, Ed., Queen Marry University, London, UK, August 2006, pp. 58–102.

[10] A. Mottaghi, K. Behdin, A. Esmaeili, M. Heydari, , and F. Marvasti, "OBTAIN: Real-time beat tracking in audio signals index terms—onset strength signal, tempo estimation, beat onset, cumulative beat strength signal, peak detection," *International Journal of Signal Processing Systems*, pp. 123–129, 2017.

[11] J. L. Oliveira, F. Gouyon, L. G. Martins, , and L. P. Reis, "IBT: A real-time tempo and beat tracking system," in *In Proc. of the 11th Intl. Conf. on Music Information Retrieval (ISMIR)*, 2014, pp. 291–296.

[12] M. Goto., "AIST annotation for the RWC music database." in *In Proc. of the 7th Intl. Conf. on Music Information Retrieval (ISMIR)*, 2006, pp. 359–360.

[13] M. Goto and Y. Muraoka, "Real-time rhythm tracking for drumless audio signals: Chord change detection for musical decisions," *Speech Communication*, vol. 27, no. 3, pp. 311–335, 1999.

[14] M. Goto and Y. Muraok, "Music understanding at the beat level real-time beat tracking for audio signals," in *In Proceedings of IJCAI- 95 Workshop on Computational Auditory Scene Analysis*, 1995, pp. 67–75.

[15] S. Durand, J. P. Bello, B. David, , and G. Richard, "Robust downbeat tracking using an ensemble of convolutional networks," *Proceedings of the 17th International Society for Music Information Retrieval Conference, ISMIR 2016*, vol. 25, no. 1, pp. 255–261, 2017.

[16] S. Durand, J. Bello, B. D., and G. Richard, "Downbeat tracking with multiple features and deep neural networks," in *In Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, 2015.

[17] S. Durand, J. P. Bello, B. D., and G. Richard, "Feature adapted convolutional neural networks for downbeat tracking," in *In Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, 2016.

[18] B. D. Giorgi, M. Mauch, , and M. Levy, "Downbeat tracking with tempo-invariant convolutional neural networks," in *In Proc. of the 17th Intl. Conf. on Music Information Retrieval (ISMIR)*, 2020, pp. 216–222.

[19] G. Peeters and H. Papadopoulos, "Simultaneous beat and downbeat-tracking using a probabilistic framework: Theory and large-scale evaluation," *IEEE Transactions on audio, speech, and language processing*, vol. 19, no. 6, August 2011.

[20] S.Böck, F. Krebs, and G. Widmer, "Joint beat and downbeat tracking with recurrent neural networks," in *In Proc. of the 7th Intl. Conf. on Music Information Retrieval (ISMIR)*, 2016.

[21] F. Krebs, S. Böck, M. Dorfer, and G. Widmer, "Downbeat tracking using beat-synchronous features and recurrent neural networks," in *In Proc. of the 17th Intl. Conf. on Music Information Retrieval (ISMIR)*, 2016.

[22] 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. of the 19th Int. Society for Music Information Retrieval Conf.*, 2018.

[23] T. Cheng, S. Fukayama, and M. Goto, "Joint beat and downbeat tracking based on CRNN models and a comparison of using different context ranges in convolutional layers," in *in Proc. of the International Computer Music Conference (ICMC)*, 2016.

[24] S. Böck and M. E. P. Davies, "Deconstruct, analyse, reconstruct: How to improve tempo, beat, and downbeat estimation," in *Proc. of the 21th Int. Society for Music Information Retrieval Conf.*, 2020, pp. 574–582.

[25] S. Böck, F. Krebs, A. Durand, S. Pöll, and R. Balsyte, "ROBOD: a real-time online beat and offbeat drummer sebastian," in *2017 IEEE signal processing cup*, 2017.

[26] C.-Y. Liang, "Implementing and adapting a downbeat tracking system for real-time applications," in *Master Thesis*, Carnegie Mellon University, 2017.

[27] S. Hainsworth and M. Macleod, "Particle filtering applied to musical tempo tracking." *EURASIP Journal on Applied Signal Processing*, vol. 15, pp. 2385–2395, 2004.

[28] S. Hainsworth and M. D. Macleod, "Beat tracking with particle filtering algorithms," in *Proc. in the IEEE Workshop on Applications of Sienal Processing to Audio and Acoustics*, 2003.

[29] A. T. Cemgil and B. Kappen, "Monte carlo methods for tempo tracking and rhythm quantization," *Journal of Artificial Intelligence Research*, vol. 18, pp. 111–222, 2003.

[30] F. Krebs, A. Holzapfel, A. T. Cemgil, and G. Widmer, "Inferring metrical structure in music using particle filters," *IEEE/ACM Transactions on audio, speech, and language processing*, vol. 23, no. 5, pp. 111–222, May 2015.

[31] M. Heydari and Z. Duan, "Don't look back: An online beat tracking method using RNN and enhanced particle filtering," in *In Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, 2021.

[32] F. Krebs, S. Böck, and G. Widmer, "An efficient state-space model for joint tempo and meter tracking," in *In Proc. of the 16th Intl. Conf. on Music Information Retrieval (ISMIR)*, 2015.

[33] F. Gouyon, A. P. 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, 2006.

[34] F. Krebs, S. Böck, and G. Widmer., "Rhythmic pattern modeling for beat and downbeat tracking in musical audio," in *Proc. of the 14th Int. Society for Music Information Retrieval Conf.*, 2013.

[35] A. Srinivasamurthy and X. Serra, "A supervised approach to hierarchical metrical cycle tracking from audio music recordings," in *In Proc. of the IEEE Int. Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2014.

[36] U. Marchand and G. Peeters, "Swing ratio estimation," in *In Proc. of the 18th Int. Conference on Digital Audio Effects (DAFx)*, 2015.

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

[38] T. de Clercq and D. Temperley., "A corpus analysis of rock harmony," *Popular Music*, vol. 30, no. 1, pp. 47–70, 2011.

[39] S. Böck, F. Korzeniowski, J. Schlüter, F. Krebs, , and G. Widmer, "Madmom: A new python audio and music signal processing library," in *in Proc. ACM Multimed. Conf, MM 2016,*, 2016, pp. 1174–1178.