Audio Engineering Society

Convention e-Brief 655

Presented at the 151st Convention
2021 October, Online

WaveBeat: End-to-end beat and downbeat tracking in the time domain

Christian J. Steinmetz and Joshua D. Reiss

Centre for Digital Music, Queen Mary University of London, UK

Correspondence should be addressed to Christian J. Steinmetz (c.j.steinmetz@qmul.ac.uk)

ABSTRACT

Deep learning approaches for beat and downbeat tracking have brought advancements. However, these approaches continue to rely on hand-crafted, subsampled spectral features as input, restricting the information available to the model. In this work, we propose WaveBeat, an end-to-end approach for joint beat and downbeat tracking operating directly on waveforms. This method forgoes engineered spectral features, and instead, produces beat and downbeat predictions directly from the waveform, the first of its kind for this task. Our model utilizes temporal convolutional networks (TCNs) operating on waveforms that achieve a very large receptive field ($\geq$ 30 s) at audio sample rates in a memory efficient manner by employing rapidly growing dilation factors with fewer layers. With a straightforward data augmentation strategy, our method outperforms previous state-of-the-art methods on some datasets, while producing comparable results on others, demonstrating the potential for time domain approaches.

1 Introduction

Beat tracking involves estimating a sequence of time instants that reflect how a human listener may tap along with a musical piece. Downbeat tracking extends this by requiring the estimation not only of the beat locations, but specifically the locations of beats corresponding to the first beat within each bar. Such a system has applications across music signal processing including automatic transcription [1], chord recognition [2], music similarity [3], and remixing [4].

Early signal processing approaches generally utilized a two-stage pipeline composed of an onset detection function followed by a post-processing phase to determine which onsets correspond to beats, often incorporating musical knowledge [3, 5, 6, 7, 8]. In contrast, with the rise of deep learning, systems have adopted predominately data driven approaches. Recurrent networks were first shown to be successful in the beat tracking task a decade ago [9], and have now been extended through a number of iterations [10, 11]. More recently, convolutional networks have been successful, performing on par with recurrent networks with greater efficiency [12]. Other works have focused on improving performance through the design of domain-inspired features [13], multi-task learning [14, 15], or specialized architectures [16].

While these deep learning approaches have demonstrated superior performance, they continue to employ aspects of traditional techniques, namely the use of hand-crafted magnitude spectrograms as input, along with specialized post-processing. The use of these features facilitates the construction of models that consider a large context efficiently, but discards a significant amount of information from the time domain signal in the process. This discarded information may be relevant for the beat tracking task, but training models directly on audio waveforms likely requires larger models, with more compute and training data [17].

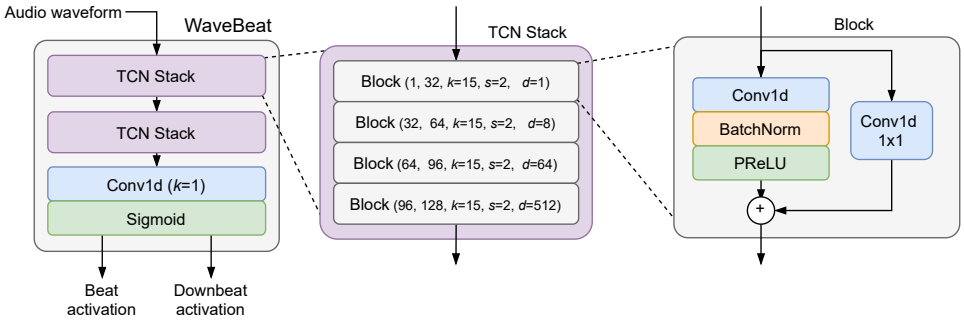

Fig. 1: The WaveBeat architecture composed of strided 1D convolutions with increasing dilation factors.

In this work, we investigate learning to jointly predict beat and downbeat events directly from raw audio. This enables us to forgo engineered features, and take advantage of information within the phase of the input signal, which has been shown to be useful in traditional approaches [18]. We also investigate if commonly employed post-processing techniques are actually required, enabling a complete end-to-end system.

We employ a specialized temporal convolutional network (TCN) [19], also known as the feedforward WaveNet [20], that achieves a significant receptive field whilst operating on audio waveforms with the use of rapidly growing dilation factors [21]. With appropriate data augmentation, our proposed model, WaveBeat, achieves comparable results to a previous deep learning approach [15], even outperforming this approach on some datasets. This indicates not only are end-to-end models feasible for this task, they may provide a pathway for improved performance. However, while these results are promising, they indicate our model struggles to generalize to unseen data distributions, lagging behind spectrogram-based approaches.

2 Proposed Model

Estimating the location of the downbeat often requires significant context, potentially upwards of 30 seconds [22]. Constructing a model with a context window or receptive field of this size for waveforms requires attending to over 1 million timesteps, which imposes significant compute and memory cost. The challenge in constructing an efficient end-to-end network has likely been a dominating factor in the use of sub-sampled spectral features in previous deep learning beat tracking approaches. To address this challenge, our proposed model incorporates two core design elements: convolutions with rapidly growing dilation patterns and carefully designed subsampling.

2.1 Architecture

The WaveBeat architecture is based on the TCN (or feedfordward WaveNet) design with a number of modifications. The block diagram in Figure 1 demonstrates the overall structure at three different levels. Starting from the lowest level, on the right, each block is composed of residual 1-dimensional convolutional layers that incorporate batch normalization followed by a PReLU activation [23]. The center of Figure 1 shows a TCN stack, which is composed of four blocks, after which the dilation pattern is repeated, a common approach [20]. The complete model is shown on the left, composed of two TCN stacks, a 1×1 convolution to downmix to two channels, and a sigmoid function to generate the beat and downbeat activations.

TCNs commonly employ a dilation pattern such that the dilation factor at each layer $l \in \{1, 2, ..., N\}$ within a stack is given by $d_l = 2^{l-1}$, a convention likely the result of the approach introduced in [24], later popularized for audio with WaveNet [25]. To achieve an even larger receptive field, we consider utilizing a dilation pattern that grows more rapidly. Inspired by recent work in audio effect modeling [21], we consider increasing the growth such that the dilation factor at each layer is given by $d_l = 8^{l-1}$.

We also note that is not useful to produce a beat activation function at audio sample rates, as the beat annotations are likely only accurate within tens of milliseconds [7]. To address this, we downsample the signal through the depth of the network by employing strided convolutions, similar to the approach used in a TCN-based encoder applied to room acoustics analysis [26]. With 8 layers, each with stride 2, we downsample the signal by a factor of $2^8 = 256$, which, given an input sample rate of 22.05 kHz produces an output signal with a sample rate of 86 Hz, close to those of previous works, which tend to be around 100 Hz [11].

2.2 Loss function

It is common to treat the beat tracking problem as a binary classification task. This is achieved by constructing a target signal y_n with a value of 1 at each timestep n containing a beat, and 0 elsewhere. Then the model is then trained with the binary cross-entropy

$$\mathscr{L}_{\text{BCE}}(\hat{y}, y) = -\frac{1}{N}\sum_{n=1}^{N}(\hat{y}_n \log(y_n) + (1-\hat{y}_n)\log(1-y_n)),$$

where the output of the model $\hat{y}_n$ is an estimate of the likelihood of a beat at the timestep n, with the total number of timesteps N. This can lead to a class imbalance across the temporal dimension, since there are often more locations with no beat. In practice, we found this encourages the model to avoid detecting beats, since such a solution will minimize the loss due to the small number of beat activations.

To address this, we adapt the mean false error $\mathscr{L}_{\text{MFE}}$ [27], a metric introduced to handle such class imbalances. Based upon the binary cross-entropy, we compute the loss as a sum of two terms that relate to the false-positive error and false-negative error

$$\mathscr{L}_{\text{FPE}}(\hat{y}, y) = \frac{1}{|J|}\sum_{j\in J}\mathscr{L}_{\text{BCE}}(\hat{y_j}, y_j)$$

$$\mathscr{L}_{\text{FNE}}(\hat{y}, y) = \frac{1}{|K|}\sum_{k\in K}\mathscr{L}_{\text{BCE}}(\hat{y_k}, y_k)$$

$$\mathscr{L}_{\text{MFE}} = \mathscr{L}_{\text{FPE}} + \mathscr{L}_{\text{FNE}},$$

where J is the set of timesteps corresponding to negative examples (no beat), and K corresponds to the positive examples (beat). This loss function attempts to balance performance by computing the sum of the average error at locations where a beat should be present, as well as the average error where there should be no beat. This encourages the model to avoid only predicting the majority class, i.e. the absence of a beat. However, we find as the sample rate of the beat activation function is reduced, this becomes less of an issue.

3 Experiments

3.1 Datasets

In order to investigate the performance of the proposed model across a number of styles and audio sources, we consider six popular beat tracking datasets: *Beatles* [28], *Hainsworth* [29], *Ballroom* [30] [31], *RWC Popular* [32], *SMC* [33], and *GTZAN* [34, 35]. Similar to previous works, we train using four datasets (*Beatles*, *Hainsworth*, *Ballroom*, and *RWC Popular*), and evaluate using two datasets that were not seen during training (*SMC* and *GTZAN*). All audio is resampled to $f_s = 22.05\,\text{kHz}$.

3.2 Training

We train WaveBeat where each convolutional layer utilizes kernels of size 15 and a stride of 2. The number of convolutional channels begins at 32 and then increases by 32 at each layer. Combined with the rapidly growing dilation factors, this enables a receptive field of over 1 million timesteps, $\approx 47\,s$, using only 8 convolutional layers. This is comparable to previous spectrogram-based beat tracking models that achieve a receptive field of around one minute [14]. However, WaveBeat has a total of 2.9 M trainable parameters, which is an order of magnitude more than common spectrogram-based models.

We utilize Adam with an initial learning rate of $1e^{-3}$, decreasing the learning rate by a factor of 10 after the beat and downbeat F-measure has not improved on the validation set for 10 epochs. To stabilize training we apply gradient clipping when the norm of the gradients exceeds 4. All models are trained with a batch size of 16 with inputs of $2^{21} = 2097152$ samples ($\approx$1.6 min at 22.05 kHz) for a total of 100 epochs. In order to balance the influence of the datasets while training, we define a single epoch to constitute 1000 random excerpts with replacement from each dataset. Additionally, we use automatic mixed precision to decrease training time and memory consumption. To facilitate reproducibility, we have made the code for these experiments available online[1].

3.3 Data augmentation

End-to-end approaches are more expressive than their counterparts that rely upon spectral features, thus they often require significantly more training data [17]. Due to the limited music recordings with beat and downbeat annotations, we found data augmentation critical in curbing overfitting. We employ a set of fairly common data augmentations, each of which has an associated probability p of being applied to each training example during a training epoch. This includes the applications of highpass and lowpass filters with random cutoff frequencies ($p = 0.25$), random

[1] https://github.com/csteinmetz1/wavebeat

			Beat			Downbeat		
Dataset	**Size**	**Model**	F-measure	CMLt	AMLt	F-measure	CMLt	AMLt
Ballroom	5 h 57 m	Spectral TCN [15]	**0.962**	**0.947**	**0.961**	0.916	0.913	**0.960**
		WaveBeat (Peak)	0.961	0.929	0.929	0.904	0.762	0.803
		WaveBeat (DBN)	0.925	0.829	0.937	**0.953**	**0.916**	0.941
Hainsworth	3 h 19 m	Spectral TCN [15]	0.902	0.848	0.930	0.722	0.696	0.872
		WaveBeat (Peak)	0.965	0.937	0.937	0.912	0.748	0.843
		WaveBeat (DBN)	**0.973**	**0.976**	**0.976**	**0.954**	**0.886**	**0.970**
Beatles	8 h 09 m	Spectral TCN [15]	-	-	-	**0.837**	**0.742**	**0.862**
		WaveBeat (Peak)	0.887	0.733	0.790	0.689	0.327	0.585
		WaveBeat (DBN)	**0.929**	**0.894**	**0.894**	0.732	0.509	0.724
GTZAN	8 h 20 m	Spectral TCN [15]	**0.885**	**0.813**	**0.931**	**0.672**	**0.640**	**0.832**
		WaveBeat (Peak)	0.825	0.682	0.767	0.563	0.279	0.515
		WaveBeat (DBN)	0.828	0.719	0.860	0.598	0.503	0.764
SMC	2 h 25 m	Spectral TCN [15]	**0.544**	**0.443**	**0.635**	-	-	-
		WaveBeat (Peak)	0.403	0.163	0.255	-	-	-
		WaveBeat (DBN)	0.418	0.280	0.419	-	-	-

Table 1: Beat and downbeat tracking results on the held-out test sets. No examples from the *GTZAN* and *SMC* datasets were seen during training.

pitch shifting between -8 and 8 semitones ($p = 0.5$), additive white noise ($p = 0.05$), applying a tanh nonlinearity ($p = 0.2$), shifting the beat locations forward or back by a random amount between $\pm$ 70 ms ($p = 0.3$), dropping a contiguous block of audio frames and beats of no more than 10% of the input ($p = 0.05$), as well as a random phase inversion ($p = 0.5$).

3.4 Post-processing

Existing beat tracking systems generally utilize a post-processing stage which inspects the beat activation functions in order to select beat locations, commonly a dynamic Bayesian network (DBN) [11]. Ideally, an end-to-end model would be able to forgo such post-processing. We analyze the beat and downbeat activation functions from WaveBeat using first simple peak picking, selecting peaks with an amplitude greater than 0.5. We then compare against beat activations produced by further post-processing with the pre-trained DBN in the `madmom` library [36] in order to examine the performance improvement.

4 Evaluation

We split the four training datasets into train/val/test sets (80%/10%/10%). We utilize the standard distance threshold of $\pm$70 ms and report the F-measure, CMLt, and AMLt metrics [28] for both beat and downbeat tracking on the test sets in Table 1. We also show the reported scores for a recent spectrogram-based TCN model [15] as a point of comparison. However, it should be noted that their scores are the result of an 8-fold cross validation, whereas our scores are computed using a single dataset split as described above. Also, their model was trained with an additional three datasets (*SMC* [33], *HJDB* [37], and *Simac* [38]), amounting to an additional 9 hours of training data, yet we employed more extensive data augmentation. In the bottom of Table 1 we report results on the left-out datasets (*GTZAN* and *SMC*), in order to test the generalization capability.

Our results demonstrate that an end-to-end approach operating directly on waveforms can in fact achieve results on-par with current state-of-the-art approaches that employ carefully engineered input features. On the *Ballroom* dataset we find that WaveBeat achieves comparable results on the beat tracking task, but achieves an improvement of 4% in downbeat tracking. Similarly, WaveBeat achieves an improvement of over 7% and 23% on the *Hainsworth* dataset for beat and downbeat tracking, respectively. While WaveBeat produces strong results on beat tracking on the *Beatles* dataset, its performance is somewhat worse on downbeat tracking.

However, these results indicate that WaveBeat falls behind the previous approach when generalizing to out-of-distribution examples and achieves comparable, yet

lower results on the *GTZAN* dataset on both beat and downbeat tracking. On the *SMC* dataset, which contains a number of challenging pieces, WaveBeat performs clearly worse than the previous approach on beat tracking. Results on the downbeat tracking task are omitted for this dataset due to the absence of downbeat annotations.

With respect to the post-processing, we find that while the pre-trained DBN brings about a small improvement in the F-measure, our end-to-end model achieves comparable performance using simple peak picking. This contrasts with previous approaches, which often report up to 15% improvement with such post-processing [11]. Surprisingly, the results in Table 1 appear to indicate that applying the DBN actually harmed performance in the case of beat tracking on the *Ballroom* dataset.

5 Conclusion

We demonstrated the ability of an end-to-end model to learn directly from waveforms on the joint beat and downbeat tracking task. With an architecture designed to efficiently achieve a large receptive field, we find that our model is able to achieve performance on-par with state-of-the-art methods for beat and downbeat tracking on some common datasets. We additionally investigate the requirement for specialized post-processing in the task of locating beat activations and find that our model performs well without such post-processing using very simple peak picking. While these results are promising, indicating that end-to-end waveform based approaches can bring improvement over existing spectrogram-based methods, additional work is needed to improve the generalization ability of these approaches. Future work involves the addition of a more rigorous data augmentation strategy, along with the application of self-supervised learning to leverage large music corpora without annotations.

6 Acknowledgement

This work is supported by the EPSRC UKRI Centre for Doctoral Training in Artificial Intelligence and Music (EP/S022694/1).

References

[1] Dixon, S., "Automatic extraction of tempo and beat from expressive performances," *Journal of New Music Research*, 30(1), 2001.

[2] Di Giorgi, B., Zanoni, M., Sarti, A., and Tubaro, S., "Automatic chord recognition based on the probabilistic modeling of diatonic modal harmony," in *nDS*, 2013.

[3] Ellis, D. P., "Beat tracking by dynamic programming," *Journal of New Music Research*, 36(1), 2007.

[4] Veire, L. V. and De Bie, T., "From raw audio to a seamless mix: creating an automated DJ system for Drum and Bass," *EURASIP Journal on Audio, Speech, and Music Processing*, 2018.

[5] Goto, M., "An audio-based real-time beat tracking system for music with or without drum-sounds," *Journal of New Music Research*, 30(2), 2001.

[6] Laroche, J., "Efficient tempo and beat tracking in audio recordings," *Journal of the Audio Engineering Society*, 51(4), 2003.

[7] Dixon, S., "Evaluation of the audio beat tracking system BeatRoot," *Journal of New Music Research*, 36(1), 2007.

[8] Davies, M. E. and Plumbley, M. D., "Context-dependent beat tracking of musical audio," *IEEE TASLP*, 15(3), 2007.

[9] Böck, S. and Schedl, M., "Enhanced beat tracking with context-aware neural networks," in *DAFx*, 2011.

[10] Böck, S., Krebs, F., and Widmer, G., "A Multi-model Approach to Beat Tracking Considering Heterogeneous Music Styles." in *ISMIR*, 2011.

[11] Böck, S., Krebs, F., and Widmer, G., "Joint Beat and Downbeat Tracking with Recurrent Neural Networks." in *ISMIR*, 2016.

[12] Davies, M. E. P. and Böck, S., "Temporal convolutional networks for musical audio beat tracking," in *EUSIPCO*, 2019.

[13] Durand, S., Bello, J. P., David, B., and Richard, G., "Downbeat tracking with multiple features and deep neural networks," in *ICASSP*, 2015.

[14] Böck, S., Davies, M. E. P., and Knees, P., "Multi-Task Learning of Tempo and Beat: Learning One to Improve the Other." in *ISMIR*, 2019.

[15] Böck, S. and Davies, M. E. P., "Deconstruct, Analyse, Reconstruct: how to improve tempo, beat, and downbeat estimation," in *ISMIR*, 2020.

[16] Di Giorgi, B., Mauch, M., and Levy, M., "Downbeat tracking with tempo-invariant convolutional neural networks," in *ISMIR*, 2020.

[17] Pons, J., Nieto, O., Prockup, M., Schmidt, E., Ehmann, A., and Serra, X., "End-to-end learning for music audio tagging at scale," in *ISMIR*, 2018.

[18] Eck, D., "Beat tracking using an autocorrelation phase matrix," in *ICASSP*, 2007.

[19] Bai, S., Kolter, J. Z., and Koltun, V., "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," *arXiv:1803.01271*, 2018.

[20] Rethage, D., Pons, J., and Serra, X., "A WaveNet for speech denoising," in *ICASSP*, 2018.

[21] Steinmetz, C. J. and Reiss, J. D., "Efficient neural networks for real-time analog audio effect modeling," *arXiv:2102.06200*, 2021.

[22] Fuentes, M., Mcfee, B., Crayencour, H. C., Essid, S., and Bello, J. P., "A music structure informed downbeat tracking system using skip-chain conditional random fields and deep learning," in *ICASSP*, 2019.

[23] He, K., Zhang, X., Ren, S., and Sun, J., "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification," in *ICCV*, 2015.

[24] Yu, F. and Koltun, V., "Multi-Scale Context Aggregation by Dilated Convolutions," in *ICLR*, 2016.

[25] Oord, A. v. d. et al., "WaveNet: A generative model for raw audio," *arXiv:1609.03499*, 2016.

[26] Steinmetz, C. J., Ithapu, V. K., and Calamia, P., "Filtered Noise Shaping for Time Domain Room Impulse Response Estimation From Reverberant Speech," *arXiv:2107.07503*, 2021.

[27] Wang, S., Liu, W., Wu, J., Cao, L., Meng, Q., and Kennedy, P. J., "Training deep neural networks on imbalanced data sets," in *IJCNN*, 2016.

[28] Davies, M. E. P., Degara, N., and Plumbley, M. D., "Evaluation methods for musical audio beat tracking algorithms," Technical Report C4DM-TR-09-06, Queen Mary University of London, 2009.

[29] Hainsworth, S. W. and Macleod, M. D., "Particle filtering applied to musical tempo tracking," *EURASIP Journal on Advances in Signal Processing*, 2004(15), 2004.

[30] Gouyon, F. et al., "An experimental comparison of audio tempo induction algorithms," *IEEE TASLP*, 14(5), 2006.

[31] Krebs, F., Böck, S., and Widmer, G., "Rhythmic Pattern Modeling for Beat and Downbeat Tracking in Musical Audio." in *ISMIR*, 2013.

[32] Goto, M., Hashiguchi, H., Nishimura, T., and Oka, R., "RWC Music Database: Popular, Classical and Jazz Music Databases." in *ISMIR*, 2002.

[33] Holzapfel, A. et al., "Selective sampling for beat tracking evaluation," *IEEE TASLP*, 20(9), 2012.

[34] Tzanetakis, G. and Cook, P., "Musical genre classification of audio signals," *IEEE Transactions on Speech and Audio Processing*, 10(5), 2002.

[35] Marchand, U., Fresnel, Q., and Peeters, G., "GTZAN-Rhythm: Extending the GTZAN testset with beat, downbeat and swing annotations," in *ISMIR*, 2015.

[36] Böck, S., Korzeniowski, F., Schlüter, J., Krebs, F., and Widmer, G., "Madmom: A new python audio and music signal processing library," in *ACM Multimedia*, 2016.

[37] Hockman, J., Davies, M. E., and Fujinaga, I., "One in the Jungle: Downbeat Detection in Hardcore, Jungle, and Drum and Bass." in *ISMIR*, 2012.

[38] Gouyon, F., *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*, Ph.D. thesis, 2005.