MULTI-TASK LEARNING OF TEMPO AND BEAT: LEARNING ONE TO IMPROVE THE OTHER

Sebastian Böck[1,3] **Matthew E.P. Davies**[2] **Peter Knees**[3]
[1] Austrian Research Institute for Artificial Intelligence (OFAI), Vienna, Austria
[2] INESC TEC, Porto, Portugal [3] TU Wien, Vienna, Austria
sebastian.boeck@ofai.at

ABSTRACT

We propose a multi-task learning approach for simultaneous tempo estimation and beat tracking of musical audio. The system shows state-of-the-art performance for both tasks on a wide range of data, but has another fundamental advantage: due to its multi-task nature, it is not only able to exploit the mutual information of both tasks by learning a common, shared representation, but can also improve one by learning only from the other. The multi-task learning is achieved by globally aggregating the skip connections of a beat tracking system built around temporal convolutional networks, and feeding them into a tempo classification layer. The benefit of this approach is investigated by the inclusion of training data for which tempo-only annotations are available, and which is shown to provide improvements in beat tracking accuracy.

1. INTRODUCTION

By definition, the music analysis tasks of tempo estimation and beat tracking are highly interconnected. Considering the goal of a beat tracking system is to produce a sequence of time instants that reflect how a human listener might tap their foot in time to a piece of music, we understand the tempo as the rate at which these beats occur, as measured in beats per minute (BPM). With the exception of a specific class of musical recordings which are both perfectly quantised (i.e. adhering strictly to a fixed metronome), and which begin precisely at the onset of a beat, e.g. drum loops, tempo information alone is insufficient to derive the beats since it provides no information about phase. In practice, a more flexible and musically realistic approach to beat tracking is required to contend with deviations from purely isochronous beat sequences without a trivial phase component. These deviations can take the form of continuous changes in tempo and/or timing which are common in expressive musical performances, more abrupt "step" changes in tempo, or short pauses after which a previously established tempo is resumed [21]. The presence and extent of these deviations from isochrony have been identified as contributing to the difficulty of musical examples for computational beat tracking [14] as well as for human annotators annotating ground truth [27].

When reflecting on the history of computational approaches for beat tracking, we consider that the role and usage of data has fundamentally changed. For the earliest work in beat tracking in the 1990s [18, 37], annotated data was scarce. By the mid-to-late 2000s, several beat tracking datasets (both public and private) came into use [12, 19, 20, 22, 24, 29] and were widely adopted as the primary means for reporting beat tracking performance. Even allowing for parameter optimisation or some training to maximise the performance of beat tracking algorithms on given datasets, a closed loop (in a strict end-to-end sense) did not exist between annotated data and beat tracking algorithms until the advent of deep neural network (DNN) approaches [7]. Both the high learning power and explicit use of annotations of DNN approaches led to a significant leap in the state of the art.

Similarly, tempo induction algorithms formerly tried to identify the main periodicity of musical accent features, such as band-passed signals, discrete onsets or a continuous detection function by means of auto-correlation [1, 13, 36], resonating comb filters [29, 37] or Fourier analysis [9], and available data was only used to evaluate the algorithms. The first attempts to learn something meaningful from data for tempo estimation sought to devise ways to choose among multiple tempo hypotheses [15, 16, 26, 38, 45] or to predict the perceptual tempo [35]. Only recently, DNN approaches have been used to infer tempo directly from spectral features [40].

At the present time, DNN approaches are highly prevalent among music analysis and generation research within the music information retrieval (MIR) community, and thus access to large amounts of high-quality annotated data is of paramount importance for the development and training of new models. For beat tracking, the hand annotation of beat locations is particularly arduous due to the need to make several hundred temporally-dependent annotations per full piece of music, and the work-load only increases in the presence of challenging musical and signal conditions [27]. By contrast, global tempo annotation, while still dependent on some approximate beat marking, can typically be created with far less effort. As a result,

© Sebastian Böck, Matthew E.P. Davies, Peter Knees. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Sebastian Böck, Matthew E.P. Davies, Peter Knees. "Multi-task Learning of Tempo and Beat: Learning One to Improve the Other", 20th International Society for Music Information Retrieval Conference, Delft, The Netherlands, 2019.

there is a far greater amount of tempo annotated data available than for beat tracking.

Our motivation is therefore towards a new approach for beat tracking which can be trained not only on beat annotations but also from tempo-only annotated data. We formulate this as a multi-task learning problem [8] for simultaneous tempo estimation and beat tracking. Our hypothesis is that due to the multi-task nature, we can not only exploit the mutual information of both tasks by learning a common, shared representation, but also improve one by learning only from the other.

We implement our multi-task approach by extending a recent beat tracking system [11] built around temporal convolutional networks (TCNs) [2, 44]. The primary addition in this paper takes the form of globally aggregating the skip connections of the TCN and feeding them into a tempo classification layer. A graphical overview of the inputs and outputs of our system is shown in Figure 1, with details of the architecture in Figure 2.

We evaluate our proposed multi-task system on a wide range of existing beat- and tempo-annotated datasets and compare performance against reference systems in both tasks. Our results demonstrate that the multi-task formulation achieves state-of-the-art performance in both tempo estimation and beat tracking. The most notable increase in performance occurs on a dataset where the network has been trained on tempo labels but whose beat annotations remain totally unseen by the network.

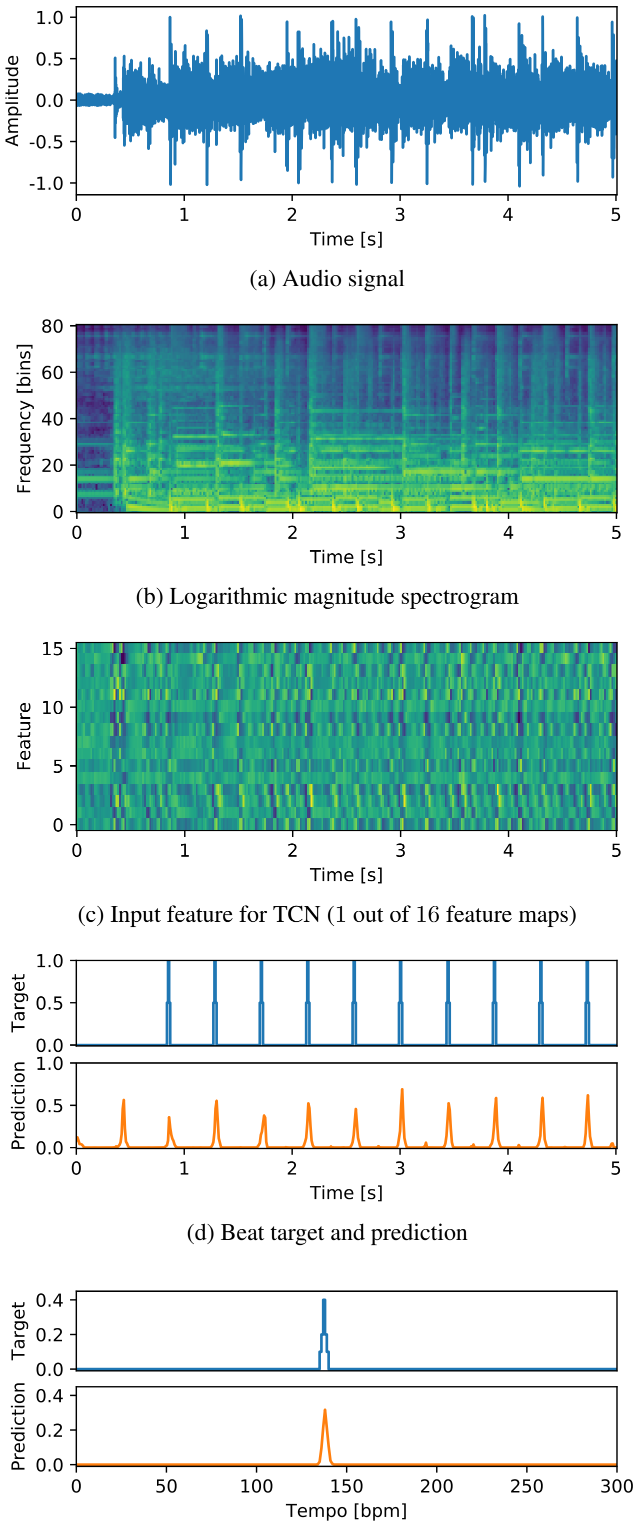
The remainder of this paper is structured as follows. In Section 2 we provide an overview of the existing beat tracking approach and then detail our multi-task formulation. In Section 3 we conduct a rigorous evaluation of beat tracking and tempo estimation. Finally, in Section 4 we discuss the context of the results and propose areas for future work.

2. APPROACH

In this section, we provide an overview of the beat tracking system [11] around which our multi-task learning approach is formulated. Following this, we describe the extension for multi-task learning via the inclusion of an additional output layer which performs tempo classification.

2.1 Beat Tracking Approach

The underlying beat tracking approach is inspired by two well-known deep learning methods: i) the *WaveNet* model [44] which uses dilated convolutions for generative audio synthesis by learning directly on raw audio waveforms, and ii) the current state of the art in musical audio beat tracking [4, 6], which uses a bi-directional long short-term memory (BLSTM) recurrent architecture. Based on the work of Bai et al. [2], who demonstrated improved performance of TCNs over recurrent architectures for numerous sequential data analysis and classification tasks, we developed a TCN approach for musical audio beat tracking [11] which, at a high-level, addressed the substitution of the BLSTM in [4, 6] with a TCN. However, since the



(a) Audio signal

(b) Logarithmic magnitude spectrogram

(c) Input feature for TCN (1 out of 16 feature maps)

(d) Beat target and prediction

(e) Tempo target and prediction

Figure 1: Signal flow of a 5 second audio excerpt through the proposed multi-task system. From the time domain signal (a), a logarithmic magnitude spectrogram is computed (b). This input representation is processed by intermediate convolutional and max pooling layers to obtain a single 16-dimensional feature (c), which is fed into the TCN. Both targets and predictions for beats and tempo are shown in (d) and (e), respectively.

TCN from *WaveNet* is both causal and operates on raw audio, several modifications were required, which are summarised below.

Instead of using raw audio as input, the dilated convolutions are performed on a highly sub-sampled low-dimensional feature representation (cf. Figure 1c). This 16-dimensional feature vector is derived by applying multiple convolution and max pooling operations to a log magnitude spectrogram of the input audio signal. The spectrogram is computed with a window and FFT size of 2048 samples, a hop size of 441 samples (i.e. 100 frames per second for audio sampled at 44100 Hz), and filtered with a bank of overlapping triangular filters with 12 bands per octave covering a frequency range of 30 to $17,000$ Hz (cf. Figure 1b). Alternating convolutional and max pooling layers are applied to slices of 5 frames in length to reduce the dimensionality both in time and frequency to a single dimension. The convolutional layers contain 16 filters each, with kernel sizes of 3×3 for the first two, and 1×8 for the last layer. The intermediate max pooling layers apply pooling only in the frequency direction over 3 frequency bins. A dropout [42] rate of 0.1 is used with the exponential linear unit (*ELU*) [10] as activation function.

The main TCN component from *WaveNet* was modified to operate non-causally, meaning that, for any time frame of the input representation, the dilated convolutions extend in both directions (i.e. back to the past and forward to the future). This provides a receptive field which is centred on the time frame in question, rather than directed solely towards the past.

In terms of the parameterisation of the TCN approach we used 11 layers with 16 1-dimensional filters of size 5 and geometrically spaced dilations ranging from 2^0 up to 2^{10} time frames. The resulting receptive field is ~ 81.5 seconds. We applied spatial dropout with rate 0.1 and used the *ELU* activation function instead of the gated activations of *WaveNet*. As output we used a single *sigmoid* unit. In order to obtain a final beat tracking output, the beat activation function produced by the network was passed to a dynamic Bayesian network, approximated by a hidden Markov model, from [31]. For further details on the TCN approach for beat tracking, see [11].

In this work we slightly changed the architecture of [11] by adding another 1×1 convolution layer with 16 filters into the residual path of the TCN layers (cf. Figure 2). We found that this layer helped to increase tempo estimation performance.

2.2 Multi-Task Extension

We extend this beat tracking system to be able to estimate the tempo of a musical piece by adding a second output branch to the network. As output, a classification layer with linear spacing as in [40] is used. It has 300 units, representing a tempo range from 0 (indicating that the piece has no tempo) up to 300 BPM. This additional output allows for multi-task learning of the whole system, the details of which are outlined in Figure 2. In order to be able to process input sequences of variable length, global average pooling (over time) is used to aggregate the features for the tempo classification layer.

While it is possible to feed the output of the TCN (or indeed the output of any other sequential beat tracking model) directly to the tempo classification layer, in practice we found that using a beat activation function led to reasonable “coarse” tempo estimation performance (i.e. determining whether a musical piece is either fast or slow), but lacked absolute precision. However, utilising skip connections of the TCN boosted tempo estimation accuracy considerably. Our intuition is that this way the subtleties of the intermediate representation of the dilated convolutions (which represent different time scales) are preserved and can be better exploited.

In the original *WaveNet* [44], skip connections were used to speed up convergence and enable training of deep models. Since the TCN beat tracking system [11] has only 11 layers, skip connections were not needed to successfully train a model and thus were not utilised. In this work, we branch off the skip connections at the same location as in *WaveNet* (i.e. from the 1×1 convolutions inside the TCN layers), but use them solely for the tempo branch of the network (cf. Figure 2).

We aggregate the skip connections of the individual layers by summation. Since the 1×1 convolutions have 16 filters each, this results in a single 16-dimensional feature vector for classification. Compared to concatenating the skip connections, this low-dimensional input to the tempo classification layer reliably prevents over-fitting to the training data. We apply dropout [41] with rate 0.5 before feeding this vector in the final tempo classification layer with a *softmax* function. During inference, quadratic interpolation of the output probability distribution is used to determine the final tempo in BPM.

The whole system has only 29,901 trainable parameters, from which the multi-task tempo classification extension accounts for 5,100. We contrast our compact model with the reported $2.9M$ parameters of the current state of the art in tempo estimation [40].

2.3 Network Training

To train the system, we represent annotated beat training data as impulse trains at the same temporal resolution as our input feature (i.e. 100 frames per second). To allow for slight deviations of the annotated beat locations and partially address the imbalance between the number of beat and non-beat frames, we use the neighbouring frames of the annotated beat positions as positive examples, but weight them by a scaling factor of 0.5 (cf. Figure 1d).

Given beat annotations, we derive tempo annotations by counting the inter-beat-intervals (IBI) to build a histogram. We smooth this histogram with a Hamming window of size 15 frames (i.e. 150 ms) to counteract small fluctuations of the beat annotations and determine the most dominant IBI by quadratic interpolation. This interval is then converted to tempo in BPM and mapped to tempo targets representing integer BPM values. In a similar way to the widening of the beat annotations, we smooth the tempo targets, but

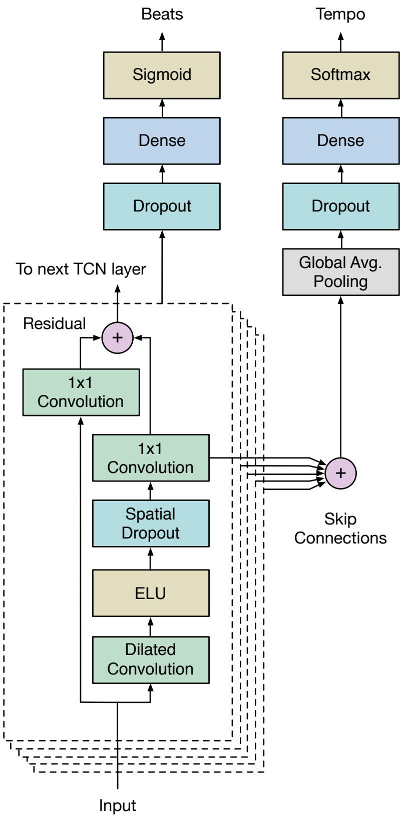


Figure 2: Structure of the neural network with the TCN for beat tracking (left) and the multi-task extension for tempo estimation (right).

extend the range to ±2 BPM, weighting the neighbouring BPM targets with 0.5 and 0.25, respectively. We then normalise the tempo targets to form a probability distribution (as shown in Figure 1e) in order for it to be usable with the *softmax* activation function.

For training, we combine the cross-entropy losses of both network outputs by weighting them equally. Since the training sequences have different lengths, we train on whole sequences and minimise the combined loss with stochastic gradient descent (i.e. using a batch size of 1). We use *Adam* [28] with an initial learn rate of 0.002, and reduce it by a factor of 5 whenever the validation loss reaches a plateau and stop training if no improvement in validation loss is observed for 50 consecutive epochs or if a maximum of 150 epochs have elapsed. To avoid exploding gradients, we clip the gradients to a maximum norm of 0.5. If only tempo targets are present for training, we mask the loss of the beat tracking output. This way, only the error of the tempo output is backpropagated through the network and used to update the weights. It is important to note, that even in this scenario the shared beat and tempo feature representation gets adapted and optimised.

3. EXPERIMENTS AND EVALUATION

For experiments and evaluation we use the datasets listed in Table 1. Those listed in the upper part are used for training using 8-fold cross validation, and those in the lower part are independent test sets held back for evaluation only. If available, updated annotations are used and indicated by additional references. We chose these datasets in order to be able to compare the performance of our proposed systems to the best performing reference systems for both beat tracking and tempo estimation.

Dataset	files	length
Ballroom [23,32] [1]	685	5 h 57 m
Beatles [12]	180	8 h 09 m
Hainsworth [24]	222	3 h 19 m
Simac [20]	595	3 h 18 m
SMC [27]	217	2 h 25 m
HJDB [25] $*$	235	3 h 19 m
ACM Mirum [35] $\star$	1410	15 h 05 m
GiantSteps [30,39] $\star$	664	22 h 05 m
GTZAN [33,43]	999	8 h 20 m

Table 1: Datasets used for training (upper half), and testing (lower half). The $*$ symbol denotes that only tempo annotations were used during training and beat annotations are used for evaluation only, and the $\star$ symbol indicates those datasets for which only tempo annotations exist.

The *HJDB* (Hardcore, Jungle, Drum & Bass) dataset is used to demonstrate the effectiveness of our multi-task extension w.r.t. its ability to improve beat tracking performance using only the tempo annotations of this set. This dataset was chosen, since its distinct music style is not well represented within any of the other training sets.

3.1 Beat Tracking Evaluation

We compare our proposed multi-task system to existing state-of-the-art beat tracking systems, namely to the underlying TCN approach presented in [11], and the two BLSTM approaches for beat [4] and joint beat and downbeat tracking [6]. Our goal is that the inclusion of the tempo classification layer is never detrimental to the performance of the beat tracking component.

Following the *de facto* standard for beat tracking evaluation, we report a set of different metrics with the parameterisation defined in [12]. We use the standard *F-measure*, as well as the continuity based measures *CMLc* and *CMLt* which require the beats to be tracked at the correct metrical level, as well as *AMLc* and *AMLt* which also allow alternate metrical interpretations such as double/half and offbeat. They either consider only the longest consecutive correctly tracked segment (*xMLc*) or all correctly tracked beats of a musical piece (*xMLt*).

From the results given in Table 2 it can be seen that all systems achieve essentially the same level of beat tracking accuracy, independent of the evaluation method. There are, however, smaller deviations from this general tendency. The beat output of the downbeat tracking system [6] performs slightly better on the *Ballroom* set, which might be

[1] The 13 identified duplicates were removed: http://media.aau.dk/null_space_pursuits/2014/01/ballroom-dataset.html

due to the characteristic rhythmic patterns which can be better exploited by explicit modelling of whole bars.

	F	*CMLc*	*CMLt*	*AMLc*	*AMLt*
			Ballroom		
BLSTM [4]	0.917	0.832	0.849	0.905	0.926
BLSTM [6]	0.938	0.872	0.892	0.932	0.953
TCN [11]	0.933	0.864	0.881	0.909	0.929
Multi-task	0.931	0.864	0.883	0.908	0.930
			Hainsworth		
BLSTM [4]	0.884	0.769	0.808	0.873	0.916
BLSTM [6]	0.871	0.732	0.784	0.849	0.910
TCN [11]	0.874	0.755	0.795	0.882	0.930
Multi-task	0.877	0.756	0.798	0.880	0.928
			SMC		
BLSTM [4]	0.529	0.296	0.428	0.383	0.567
BLSTM [6]	0.516	0.307	0.406	0.429	0.575
TCN [11]	0.543	0.315	0.432	0.462	0.632
Multi-task	0.535	0.295	0.415	0.440	0.613
			GTZAN		
BLSTM [4]	0.864	0.750	0.768	0.901	0.927
BLSTM [6]	0.856	0.716	0.744	0.876	0.919
TCN [11]	0.843	0.695	0.715	0.889	0.914
Multi-task	0.847	0.702	0.724	0.886	0.916

Table 2: Beat tracking results on datasets used for training with 8-fold cross validation (top), and on completely unseen test data (bottom).

Given these results, we infer that the multi-task system achieves at least the same performance as the same system without the multi-task extension.

3.2 Multi-Task Evaluation

In the previous section, our evaluation focused on the use of both tempo- and beat-annotated training data within our multi-task model. In order to test our hypothesis that tempo-only information can indeed lead to improved beat tracking accuracy, and thus demonstrate the ability of multi-task learning to strengthen one target by learning additionally from the other, we perform a further experiment. To this end, we add a new dataset, but only use its tempo annotations for training.

We believe that the effect of this learning strategy should be most visible when performed with data, which is otherwise underrepresented in the training set. In our opinion, the *HJDB* dataset is a perfect fit since it contains musical genres from the early 1990s, namely Hardcore, Jungle, and Drum & Bass, which are characterised by their very distinct rhythmic structure. For the details on this dataset, see [25].

We train our new multi-task approach in two different ways. Once with the data as outlined in Table 1, but without *HJDB* (i.e. as in the previous section), and once including the tempo annotations of this dataset.

Inspection of the first two rows of Table 3 reveals that both the original TCN beat tracking system, and the system with the multi-task extension achieve roughly the same performance across all evaluation methods. However, once the additional tempo information is utilised (last row marked with the $*$ symbol), the performance increases by up to ~ 5 percentage points. The jump in accuracy is best observed in the *CMLc* and *CMLt* evaluation methods. This indicates that the system is able to exploit the additional information to track the beats at the correct metrical level more often than without this information. Within the context of the *HJDB* dataset where the “correct” metrical level is largely unambiguous, we consider this to be an important contribution.

	F	*CMLc*	*CMLt*	*AMLc*	*AMLt*
			HJDB		
TCN [11]	0.842	0.802	0.810	0.903	0.912
Multi-task	0.850	0.800	0.804	0.921	0.927
Multi-task $*$	0.882	0.848	0.858	0.937	0.947

Table 3: Multi-task learning beat tracking results on the *HJDB* dataset. All results obtained with 8-fold cross validation. The $*$ symbol denotes that tempo annotations of the *HJDB* set were used as additional targets during training.

3.3 Tempo Evaluation

Further to the beat tracking oriented evaluation results reported in the previous two sections, we also explore the effectiveness of our proposed approach for the task of global tempo estimation. To discover how our multi-task approach compares to the state of the art, we contrast its performance against four reference systems [5, 17, 36, 40]. Following the established evaluation practice for tempo estimation [23] we report the *Accuracy 1* and *Accuracy 2* scores with a tolerance of $\pm 4\%$ for each of these methods, with the results shown in Table 4.

Given that human perception of tempo is known to be subjective [34], this very reasonably manifests in multiple, valid interpretations of the beat among listeners and thus more than one acceptable tempo. Thus, in the context of automatic tempo estimation, it may not be realistic to expect to obtain near perfect performance on the *Accuracy 1* score on datasets of arbitrary musical makeup. To this end, we rely on the *Accuracy 2* score (which permits so-called “tempo octave errors”) to better gauge performance.

On all of the reported datasets in Table 4, our proposed approach is the only one to consistently obtain an *Accuracy 2* greater than or equal to 0.938, which shows the high potential of our method to accurately find tempo across diverse musical data. Even with the stricter *Accuracy 1* evaluation, our method achieves at least a score of 0.697 which is ahead of all other methods, albeit by a small margin. It is important to stress that the *ACM Mirum*, *GiantSteps*, and *GTZAN* datasets are completely unseen by our multi-task approach, and this pattern even holds for *HJDB* when not included in the training set.

Concerning the *HJDB* set, we can observe a different overall pattern of performance compared to the other datasets, with a much smaller gap between *Accuracy 1* and

	Accuracy 1	*Accuracy 2*
ACM Mirum		
Gkiokas et al. [17]	0.725	0.979
Percival and Tzanetakis [36]	0.733	0.972
Böck et al. [5]	0.741	0.976
Schreiber and Müller [40]	0.795	0.974
Multi-task	0.757	0.977
Multi-task ∗	0.749	0.974
GiantSteps		
Gkiokas et al. [17]	0.721	0.922
Percival and Tzanetakis [36]	0.506	0.956
Böck et al. [5]	0.589	0.864
Schreiber and Müller [40]	0.730	0.893
Multi-task	0.697	0.958
Multi-task ∗	0.764	0.958
GTZAN		
Gkiokas et al. [17]	0.651	0.931
Percival and Tzanetakis [36]	0.658	0.924
Böck et al. [5]	0.697	0.950
Schreiber and Müller [40]	0.694	0.926
Multi-task	0.697	0.939
Multi-task ∗	0.673	0.938
HJDB		
Gkiokas et al. [17]	0.783	0.911
Percival and Tzanetakis [36]	0.285	1.0
Böck et al. [5]	0.796	0.868
Schreiber and Müller [40]	0.902	0.991
Multi-task	0.826	0.962
Multi-task ∗ †	1.0	1.0

Table 4: Tempo estimation results on completely unseen data. The ∗ symbol denotes that tempo annotations of the *HJDB* set were used as additional targets during training, the † symbol results obtained with 8-fold cross validation.

Accuracy 2 for most systems. Echoing the situation in the beat tracking evaluation in Table 3, we believe that this is a direct result of the unambiguous tempo for these styles of music. Looking across the performance of the other algorithms on *HJDB*, we discover that the method of Percival and Tzanetakis [36], while it also obtains a perfect score for *Accuracy 2*, is largely unable to identify the annotated tempo as shown by the disproportionately low score for *Accuracy 1*.

When trained with the additional tempo annotations of the *HJDB* set, our multi-task method is the only one able to detect the correct tempo for all pieces of this dataset for both *Accuracy 1* and then trivially for *Accuracy 2*. Although results are obtained with cross-validation, this was to be expected because of the homogeneity of the dataset. *Accuracy 1* on the *GiantSteps* set also greatly benefits from this additional training material, since this dataset contains a huge proportion of music labelled with the musical genre "drum and bass". On the other hand, having access to this kind of data (the system of Schreiber and Müller [40] was trained on an extended version of the *GiantSteps* dataset) can in turn result in very good scores on the *HJDB* set.

4. DISCUSSION AND CONCLUSIONS

In this paper, we have proposed a novel formulation for the simultaneous estimation of tempo and beat from musical audio signals within a multi-task learning framework. Via an extensive evaluation of both beat tracking and tempo estimation, we have demonstrated that our proposed multi-task approach leads to state-of-the-art performance across a wide variety of test datasets and relevant evaluation methods. Perhaps most critically, we have shown that, within this multi-task learning framework, we can improve the performance of beat tracking by providing it tempo-only annotations. In light of the challenges of obtaining high-quality annotated data for training beat tracking systems, the ability to profit from alternative training data which is both far more prevalent and easier to annotate, may have a significant impact on beat tracking moving forward.

In order to train our model, we made use of all of the available beat and tempo annotations within the allocated training sets in Table 1, and subsequently provided additional tempo-only annotations for evaluation on the *HJDB* dataset. We consider this split between beat and tempo annotated data to be one that is worthy of further exploration, in particular by seeking to understand how little beat annotated data is sufficient to achieve the same performance, assuming we can supplement the model with additional tempo annotations. This reduction of beat information could be posed in two ways, either by a lower number of fully annotated excerpts/pieces, or by restricting the duration of annotated sections across many pieces. If successful, the latter option would offer the possibility to rapidly increase the availability of training data by drastically reducing the burden of annotating long pieces of music—at least for those with roughly constant tempo.

We frame this discussion within the computational context of our proposed multi-task approach and the TCN beat tracker [11] which it extends. As previously stated, our multi-task model is highly effective in terms of objective performance, but with a fraction of the number of weights of other state-of-the-art approaches. This has two particularly beneficial properties. First, it allows for very efficient training (thanks in part to the ease of parallelisation of dilated convolutional models compared to recurrent architectures). Second, the training of networks with very few weights drastically reduces the degrees of freedom of the network, and hence strongly mitigates over-fitting. Thus, when looking beyond the limited domain of existing annotated datasets and considering generalisation capabilities of beat tracking and tempo estimation methods (and the subsequent re-use of this information for end-users) on totally unseen data, we believe that such "compact" deep models are worthy candidates for future research.

Supplementary material can be found online at `https://github.com/superbock/ISMIR2019` with executable code and pre-trained models being included in *madmom* [3] (`https://github.com/CPJKU/madmom`).

5. ACKNOWLEDGEMENTS

Sebastian Böck is supported by the Austrian Promotion Agency (FFG) under the "BASIS, Basisprogramm" umbrella program. Matthew E.P. Davies is supported by Portuguese National Funds through the FCT-Foundation for Science and Technology, I.P., under the project IF/01566/2015.

6. REFERENCES

[1] M. Alonso, G. Richard, and B. David. Accurate tempo estimation based on harmonic + noise decomposition. *EURASIP Journal on Applied Signal Processing*, pages 161–161, 2007.

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

[3] S. Böck, F. Korzeniowski, J. Schlüter, F. Krebs, and G. Widmer. Madmom: A new python audio and music signal processing library. In *Proc. of the 2016 ACM Multimedia Conf.*, pages 1174–1178, 2016.

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

[5] S. Böck, F. Krebs, and G. Widmer. Accurate tempo estimation based on recurrent neural networks and resonating comb filters. In *Proc. of the 16th Intl. Society for Music Information Retrieval Conf. (ISMIR)*, pages 625–631, 2015.

[6] S. Böck, F. Krebs, and G. Widmer. Joint beat and downbeat tracking with recurrent neural networks. In *Proc. of the 17th Intl. Society for Music Information Retrieval Conf. (ISMIR)*, pages 255–261, 2016.

[7] S. Böck and M. Schedl. Enhanced beat tracking with context-aware neural networks. In *Proc. of the 14th Intl. Conf. on Digital Audio Effects (DAFx)*, pages 135–139, 2011.

[8] R. Caruana. Multitask learning. *Machine Learning*, 28:41–75, 1997.

[9] A. T. Cemgil, B. Kappen, P. Desain, and H. Honing. On tempo tracking: Tempogram representation and Kalman filtering. *Journal of New Music Research*, 28:4:259–273, 2001.

[10] D. Clevert, T. Unterthiner, and S. Hochreiter. Fast and accurate deep network learning by exponential linear units (ELUs). In *Proc. of the 4th Intl. Conf. on Learning Representations (ICLR)*, 2016.

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

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

[13] S. Dixon. Automatic extraction of tempo and beat from expressive performances. *Journal of New Music Research*, 30:39–58, 2001.

[14] S. Dixon. An empirical comparison of tempo trackers. In *Proc. of the 8th Brazilian Symp. on Computer Music*, pages 832–840, 2001.

[15] A. Elowsson. Beat tracking with a cepstroid invariant neural network. In *Proc. of the 17th Intl. Society for Music Information Retrieval Conf. (ISMIR)*, pages 351–357, 2016.

[16] A. Gkiokas, V. Katsouros, and G. Carayannis. Reducing tempo octave errors by periodicity vector coding and SVM learning. In *Proc. of the 13th Intl Society for Music Information Retrieval Conf. (ISMIR)*, pages 301–306, 2012.

[17] A. Gkiokas, V. Katsouros, G. Carayannis, and T. Stafylakis. Music tempo estimation and beat tracking by applying source separation and metrical relations. In *Proc. of the 37th IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, pages 421–424, 2012.

[18] M. Goto and Y. Muraoka. A beat tracking system for acoustic signals of music. In *Proc. of the 2nd ACM Intl. Conf. on Multimedia*, pages 365–372, 1994.

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

[20] 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.

[21] F. Gouyon and S. Dixon. A review of automatic rhythm description systems. *Computer Music Journal*, 25(1):34–54, 2005.

[22] F. Gouyon and P. Herrera. Determination of the meter of musical audio signals: Seeking recurrences in beat segment descriptors. In *Audio Engineering Society Convention 114*, 2003.

[23] 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*, 14(5):1832–1844, 2006.

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

[25] J. Hockman, M. E. P. Davies, and I. Fujinaga. One in the Jungle: Downbeat detection in Hardcore, Jungle, and Drum and Bass. In *Proc. of the 13th Intl. Society for Music Information Retrieval Conf. (ISMIR)*, pages 169–174, 2012.

[26] J. Hockman and I. Fujinaga. Fast vs slow: Learning tempo octaves from user data. In *Proc. of the 11th Intl. Society for Music Information Retrieval Conf. (ISMIR)*, pages 231–236, 2010.

[27] A. Holzapfel, M. E. P. Davies, J. R. Zapata, J. L. Oliveira, and F. Gouyon. Selective sampling for beat tracking evaluation. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(9):2539–2548, 2012.

[28] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *Proc. of the 3rd Intl. Conf. for Learning Representations (ICLR)*, 2015.

[29] A. Klapuri, A. Eronen, and J. Astola. Analysis of the meter of acoustic musical signals. *IEEE Transactions Speech and Audio Processing*, 14(1):342–355, 2006.

[30] P. Knees, A. Faraldo, P. Herrera, R. Vogl, S. Böck, F. Hörschläger, and M. Le Goff. Two data sets for tempo estimation and key detection in electronic dance music annotated from user corrections. In *Proc. of the 16th Intl. Society for Music Information Retrieval Conf. (ISMIR)*, pages 364–370, 2015.

[31] F. Krebs, S. Böck, and G. Widmer. An efficient state space model for joint tempo and meter tracking. In *Proc. of the 16th Intl. Society for Music Information Retrieval Conf. (ISMIR)*, pages 72–78, 2015.

[32] F. Krebs, S. Böck, and G. Widmer. Rhythmic pattern modeling for beat and downbeat tracking in musical audio. In *Proc. of the 14th Intl. Society for Music Information Retrieval Conf. (ISMIR)*, pages 227–232, 2013.

[33] U. Marchand and G. Peeters. Swing ratio estimation. In *Proc. of the 18th Intl. Conf. on Digital Audio Effects (DAFx)*, pages 423–428, 2015.

[34] D. Moelants and M. McKinney. Tempo perception and musical content: What makes a piece fast, slow or temporally ambiguous. In *Proc. of the 8th Intl. Conf. on Music Perception and Cognition*, pages 558–562, 2004.

[35] G. Peeters and J. Flocon-Cholet. Perceptual tempo estimation using GMM-regression. In *Proc. of the 2nd ACM workshop on music information retrieval with user-centered and multimodal strategies (MIRUM)*, pages 45–50, 2012.

[36] G. Percival and G. Tzanetakis. Streamlined tempo estimation based on autocorrelation and cross-correlation with pulses. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 22(12):1765–1776, 2014.

[37] E. D. Scheirer. Tempo and beat analysis of acoustic musical signals. *The Journal of the Acoustical Society of America*, 103(1):588–601, 1998.

[38] H. Schreiber and M. Müller. A post-processing procedure for improving music tempo estimates using supervised learning. In *Proc. of the 18th Intl. Society for Music Information Retrieval Conf. (ISMIR)*, pages 235–242, 2017.

[39] H. Schreiber and M. Müller. A crowdsourced experiment for tempo estimation of electronic dance music. In *Proc. of the 19th Intl. Society for Music Information Retrieval Conf. (ISMIR)*, pages 409–415, 2018.

[40] H. Schreiber and M. Müller. A single-step approach to musical tempo estimation using a convolutional neural network. In *Proc. of the 19th Intl. Society for Music Information Retrieval Conf. (ISMIR)*, pages 100–105, 2018.

[41] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[42] J. Tompson, R. Goroshin, A. Jain, Y. LeCun, and C. Bregler. Efficient object localization using convolutional networks. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 648–656, 2015.

[43] G. Tzanetakis and P. Cook. Musical genre classification of audio signals. *IEEE Transactions on Speech and Audio Processing*, 10(5):293–302, 2002.

[44] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu. Wavenet: A generative model for raw audio. *CoRR abs/1609.03499*, 2016.

[45] F.-H. F. Wu and J.-S. R. Jang. A supervised learning method for tempo estimation of musical audio. In *22nd Mediterranean Conf. of Control and Automation (MED)*, pages 599–604, 2014.