# MUSICAL TEMPO ESTIMATION USING A MULTI-SCALE NETWORK

**Xiaoheng Sun**[1] **Qiqi He**[1] **Yongwei Gao**[1] **Wei Li**[1,2]
[1] School of Computer Science and Technology, Fudan University, Shanghai, China
[2] Shanghai Key Laboratory of Intelligent Information Processing, Fudan University, Shanghai, China
{19210240112, heqq20, ywgao16, weili-fudan}@fudan.edu.cn

## ABSTRACT

Recently, some single-step systems without onset detection have shown their effectiveness in automatic musical tempo estimation. Following the success of these systems, in this paper we propose a Multi-scale Grouped Attention Network to further explore the potential of such methods. A multi-scale structure is introduced as the overall network architecture where information from different scales is aggregated to strengthen contextual feature learning. Furthermore, we propose a Grouped Attention Module as the key component of the network. The proposed module separates the input feature into several groups along the frequency axis, which makes it capable of capturing long-range dependencies from different frequency positions on the spectrogram. In comparison experiments, the results on public datasets show that the proposed model outperforms existing state-of-the-art methods on Accuracy1.

## 1. INTRODUCTION

Although there are many different ways to describe musical tempo (e.g., measures per minute, bars per minute, or even a range of Italian terms), beats per minute (BPM) is the most commonly used measurement unit. The estimation of BPM plays an important role in a variety of applications, such as music recommendation, automatic accompaniment, playlist generation, etc. Because of its utility, the automatic estimation of tempo has been an important task and received continuous attention in the field of music information retrieval (MIR) [1–4].

Traditional methods for automatic tempo estimation are usually based on hand-crafting signal processing. To estimate the tempo of a given audio segment, an onset strength signal (OSS) function is firstly derived, and the frequency of the major pulses is extracted and converted to BPM. The OSS function is a function whose peaks should correspond to onset times. It can be obtained by various methods, such as means of auto-correlation [5, 6], comb filters [2, 7] and Fourier analysis [8]. Machine learning techniques are also adopted for tempo estimation, including Gaussian mixture models (GMM) [9], support vector machines (SVM) [10, 11], k-nearest neighbors (k-NN) [12, 13], random forests [14] and so on. Since Böck [15] proposed a recurrent neural network (RNN) model to learn beat-level representations from audio signals, attempts to use deep neural networks (DNN) for tempo estimation began to grow [16–18].

In all methods mentioned above, the extraction of BPM depends on some post-processing of OSS functions or beat activation functions. It is only in recent years that the *single-step* tempo estimation systems based on DNN appeared. As the first single-step approach for tempo estimation, the CNN model proposed by Schreiber [19] is capable of extracting BPM value directly from a Mel-scaled spectrogram. In this work, classification is proved to be an effective method for tempo estimation. Adopting a similar idea, Foroughmand [20] proposed the Harmonic-Constant-Q-Modulation (HCQM), a new representation of audio signal, as the input of a relatively simple CNN classification model. The experimental results also showed its effectiveness.

A commonly used metric in tempo estimation is Accuracy1 [3], indicating the percentage of correct estimates allowing a $\pm 4\%$ tolerance. However, automatic tempo estimation systems tend to predict a wrong tempo by a factor of 2 or 3, known as *octave errors*. As an additional measure, Accuracy2 is introduced, which ignores octave errors. In some applicational scenarios (such as DJ software), accurate tempo annotations are mandatory and octave errors are unacceptable [21], but most existing algorithms' performance on Accuracy1 is still far from satisfactory.

Previous works [19, 20] have shown the potential of CNN-based single-step approach to improve performance on Accuracy1. Following the success of these methods, in this paper we propose a CNN-based single-step model named Multi-scale Grouped Attention Network (MGANet). A multi-scale network architecture is designed to aggregate information from different scales to produce superior feature representations. Furthermore, a Grouped Attention Module (GAModule) is proposed to capture long-range dependencies and refine the feature based on the attention mechanism.

The remainder of this paper is organized as follows. In Section 2, we introduce the proposed method in detail. In Section 3, experimental results are presented to show the effectiveness of our method. Finally, we make further conclusion in Section 4.

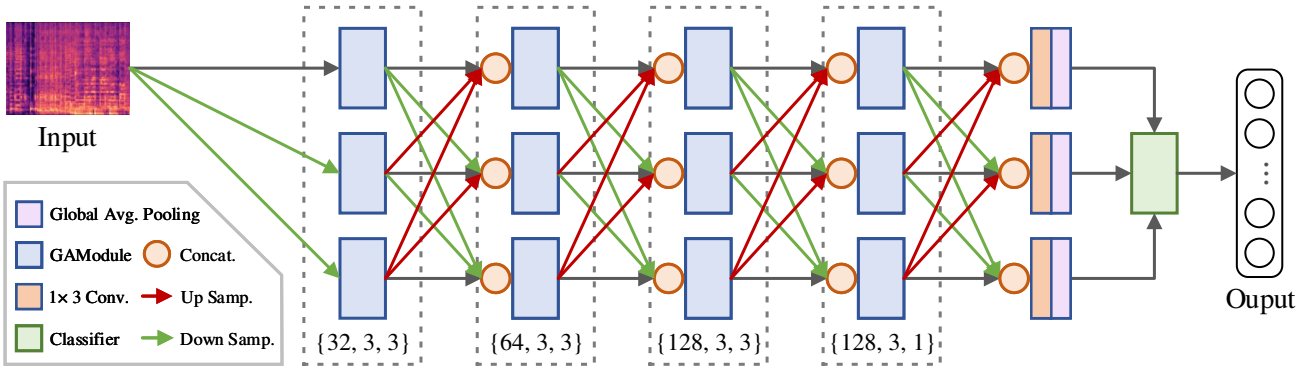



**Figure 1**: The overall architecture of Multi-scale Grouped Attention Network (MGANet). The numbers in dashed boxes indicate the three parameters of GAModules: {output channel number $C$, pooling size $p$, group number $k$}. Every concatenation operation in the figure is followed by a $1\times1$ convolution layer to adjust channel number. The classifier consists of a concatenation operation, a fully connected layer, and a softmax layer.

## 2. APPROACH

### 2.1 Proposed Model

Same as [19] and [20], we also treat tempo estimation as a classification problem. The output of our model is a probability distribution of 256 BPM classes (from 30 to 285 BPM). Because the Mel-scaled frequency matches closely the human auditory perception, we choose the Mel-scaled spectrogram as the raw feature. First, the original audio data is resampled to 11.025 kHz. Then, we use half-overlapping windows of 1,024 frames, and transform each window into an 81-band Mel-scaled magnitude spectrum. The input of the proposed model is designed as a spectrogram segment of 128 frames, roughly 6 seconds long.

In the rest of this section, we first present the overall architecture of the proposed MGANet. Then, we introduce the GAModule, which is the key component of the network.

#### *2.1.1 Multi-scale Network Architecture*

The goal of tempo estimation is to extract a periodic pattern from an audio signal. Therefore, global information of the input spectrogram is particularly important. Due to the characteristics of CNN, overall pattern extraction is usually achieved by stacking multiple layers. But directly repeating convolution layers makes the model difficult to design and optimize. Another way is to use large-size convolution kernels to enlarge the receptive fields. However, this is also costly because of the increase in parameters and multiply-add operations. To solve the problem, we introduce the idea of multi-scale structure, which has been proved to be effective in many classification tasks [22–24]. By downsampling / upsampling the feature to different scales and exchanging information repeatedly, high-level representations can be derived after just a few layers.

As shown in Figure 1, the overall architecture of MGANet is mainly composed of three branches for different scale. In each branch, input features are gradually downsampled over the frequency (vertical) axis, but maintains the resolution through the whole process on the time (lateral) axis. Furthermore, these feature maps from different scales are merged repeatedly to integrate contextual information, leading to high-level representations amenable to classification.

Specifically, the input spectrogram is first downsampled by 1/2 and 1/4 over the time axis with average pooling, resulting in three representations of sizes (81, 128), (81, 64), and (81, 32). Then, the representations are fed into three parallel branches respectively to perform feature processing. The processing is mainly done by the proposed GAModule described in section 2.1.2. Through the whole structure, we repeat multi-scale fusion by rescaling and concatenation. Average pooling and transposed convolution [25] layers with kernel size of $1 \times 3$ are used to perform rescaling. For concatenation, a $1 \times 1$ convolution layer with the exponential linear unit (ELU) [26] activation is followed to adjust the channel number.

Processed by GAModules, the features are gradually downsampled over the frequency axis to summarize frequency bands, making the representations easier to detect periodicity. On each branch, the downsampling is repeated four times. Accordingly, the channel numbers of the features are increased. After the above processes, three feature maps with shapes (1, 128, 128), (1, 64, 128), and (1, 32, 128) are obtained. Then, these feature maps are fused again and fed into a $1 \times 3$ convolution layer to adjust channel numbers to 256. After global average pooling, three vectors of length 256 are concatenated together. Finally, a fully connected layer takes the vector as input and a softmax layer is used to derive the probability distribution of 256 tempo classes.

#### *2.1.2 Grouped Attention Module*

The proposed GAModule structure is shown in Figure 2. The module consists of two parts: a trunk branch performing feature processing, and $k$ attention branches producing an attention mask to capture global context information and recalibrate the output feature map.

The structure of the attention branch is mainly inspired by the global context network (GCNet) [27], which is de-

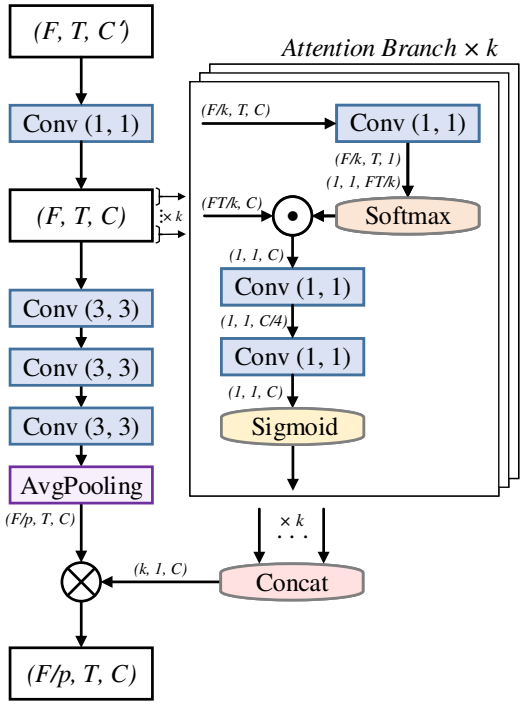



**Figure 2**: The structure of Grouped Attention Module (GAModule). Feature maps are shown as feature dimensions, e.g. $(F, T, C)$ denotes a feature map with height $F$, width $T$, and channel number $C$. $p$ and $k$ denote pooling size and group number respectively. $\odot$ denotes matrix multiplication and $\otimes$ denotes broadcast element-wise multiplication.

signed for long-range dependency modeling through attention mechanism. The attention mechanism biases the allocation of the most informative feature expressions and suppresses the less useful ones. Recently, the benefits of the attention mechanism have been demonstrated in a series of tasks. We introduce the attention mechanism into GAModule mainly for two purposes: 1) model the long-range dependencies to obtain global context features; 2) reweight the importance of different channels to improve the representational capacity of the refined feature.

Unlike the images in the field of computer vision, the two axes of audio spectrograms have different meanings, which respectively represent frequency and time. Furthermore, it is known that different musical instruments have different frequency ranges, and different frequency ranges have a different impact on the total sound. These facts indicate that different frequency bands contain relatively independent information. Based on these observations, we believe that it's inappropriate to aggregate the whole spatial scope at once to calculate long-range dependencies. Instead, different frequency positions of the feature should be handled separately, which will help to filter the useful information more efficiently. Therefore, different from traditional channel-wise attention models that aggregate the entire feature to generate one attention map (e.g., squeeze-and-excitation networks [28]), we divide the feature equally into $k$ groups along the frequency axis and send each fragment into an independent attention branch. We termed the operation as *grouped channel attention*.

As shown in Figure 2, the framework of the attention branch is roughly the same as the GC block in GCNet. Firstly, the feature map is squeezed into a channel descriptor by global attention pooling. The pooling is achieved by convolution, softmax, and matrix multiplication. For an input feature map $x$, the generated descriptor $z \in \mathbb{R}^C$ is calculated by

$$z = \sum_{j=1}^{N_p} \frac{exp(ELU(Wx_j))}{\sum_{m=1}^{N_p} exp(ELU(Wx_m))} x_j \quad (1)$$

where $j$ and $m$ enumerate all possible positions, and $W$ denotes linear transformation matrix. We adopt ELU as the activation of the convolution layer to further increase robustness. After the pooling, global spatial information is gathered in the descriptor. Then, a bottleneck of two-layer architecture is formed to transform information. We adopt a reduction ratio of 4 and ELU activation in the first layer. A sigmoid function is then applied to rescale the transformation output. Finally, $k$ attention maps with the shape of $(1, 1, C)$ can be obtained. We concatenate these attention maps along the frequency axis and get the output attention map of $(k, 1, C)$.

Simultaneously, in the trunk branch we simply stack three convolution layers with kernel of $3 \times 3$ and ELU activation. Because of the existence of attention branches, the trunk does not need a complex structure and too many layers, which reduces the number of parameters and the complexity of the model. We use average pooling with pooling size of $p \times 1$ to downsample the feature map to $(F/p, T, C)$. Finally, broadcast element-wise multiplication is performed to fuse the output of the trunk branch and attention branches. Through the fusion, the output feature map is refined by global contextual information gathered by grouped attention operation.

### 2.2 Training Data & Augmentation

For training and validation, we adopt the three training datasets used in [19]: *LMD Tempo* (3,611 items), *MTG Tempo* (1,159 items), and *Extended Ballroom* (3,826 items). However, though covering multiple musical genres, the combination of these datasets is not genre-balanced, and some common genres are even missing. It is known that tempo perception is closely related to music genre. For example, for popular music, people usually perceive tempo through drumbeats, while for classical music, people often perceive tempo from bass instruments such as double bass. To alleviate the genre imbalance, we use two additional datasets to supplement the training data:

- ***RWC-popular***: To further enhance the model's ability to estimate pop music tempo, we used RWC-popular [29] (a pop music database with 100 pieces) for training. We cut the songs into 30s fragments without overlapping, resulting in 735 items.

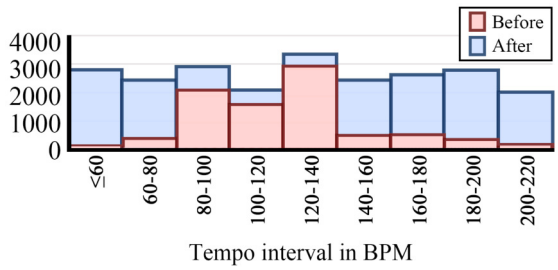



**Figure 3**: Tempo distribution before and after augmentation.

- ***FD-Tempo***: To enrich the genres of training data, we selected some tracks of classical music. For each track, we chose several 30s excerpts with stable tempi and annotated them by manually tagging. Finally, 530 items are obtained as an additional dataset termed *FD-Tempo*.

We use the combination of the five datasets for training and validation. It contains 9,861 tracks with a total length of 41h 3min. Specifically, we randomly choose 500 tracks for validation, and the rest 9,361 tracks are used for training.

To alleviate the BPM class imbalance, we further augment the training set by speeding up / slowing down the selected tracks with factors randomly chosen from 0.7∼1.4 without altering the pitch. We retain the original files and make sure that the same audio will not be selected more than 15 times. After augmentation, the number of tracks increases from 9,361 to 23,512. Note that the validation set is not augmented. The tempo distribution in the training set before and after augmentation is shown in Figure 3. Besides, we also adopt the scale-&-crop data augmentation mentioned in [19] to further increase the variability of training data.

### 2.3 Training Details

For training, the batch size we set is 32. In each epoch, 128 consecutive frames of each sample are randomly selected for training. We choose the categorical cross-entropy as the loss function, and an Adam optimizer [30] is applied with a learning rate of 0.001. We evaluate Accuracy1 of the validation set every 500 iterations, and save the model with the highest accuracy. The training is not stopped until Accuracy1 has not improved for 50,000 iterations.

## 3. EVALUATION

We choose Accuracy1 (ACC1) and Accuracy2 (ACC2) [3] as the evaluation metrics. Accuracy1 is defined as the percentage of correct estimates allowing a $\pm 4\%$ tolerance. Accuracy2 ignores octave errors by a factor of 2 and 3, and also allows a $\pm 4\%$ tolerance. As mentioned earlier, the demand for highly accurate tempo annotations has become increasingly urgent in many applicational scenarios. Hence we mainly focus on improving Accuracy1.

We focus on the performance on global tempo estimation based on the assumption the tempo of the input track stays constant, and only one BPM value will be returned by the estimation system. In the experiment, the global tempo is obtained by averaging the outputs of softmax layer over different parts of a full track [19].

| Method | ACC1 | ACC2 | ACC1 | ACC2 |
|---|---|---|---|---|
| w/o AB | 77.0 | 89.9 | 79.8 | 95.3 |
| w/o GA | 78.5 | 89.1 | 79.0 | 94.2 |
| Single-scale | 75.8 | 89.6 | 71.2 | 94.5 |
| Proposed | **78.9** | **91.3** | **82.1** | **95.7** |
| | (a) GTzan | | (b) ACM Mirum | |

**Table 1**: Results of ablation study. "w/o AB" and "w/o GA" denote "without attention branch" and "without grouped attention" respectively. Best results are set in **bold**.

### 3.1 Ablation Study

We study the effect of each idea in our approach. To simplify the discussion, we select two test datasets *GTzan* [31] and *ACM Mirum* [9] for analysis. These two datasets are relatively large (999 and 1,410 items respectively), and both cover rich genres.

To investigate how much the proposed GAModule contributes to the model, we design a set of experiments. Firstly, we remove the attention branches in the module, and only the trunk branch is remained to process features. As shown in Table 1, the performance degrades for both datasets. When focusing on Accuracy1, the performance decreases by 1.9% for *GTzan* and 2.3% for *ACM Mirum*. Then, in another experiment we keep only one attention branch in each module, which can be achieved by setting GAModules' parameter $k$ to 1. The Accuracy1 reduced by 0.4% and 3.1% respectively. For Accuracy2, in both experiments there is also a certain degree of decline. These results indicate that the attention mechanism is helpful to capturing long-range dependencies and therefore improve the generalization of the model. But directly using existing modules may hinder the effect. The proposed grouped attention takes into account the characteristics of spectrogram and achieves further improvements of the model.

Then, we analyze the effect of the multi-scale architecture by changing the architecture to a single-scale one. We remove all downsampled subnetworks and only retain the one with the highest resolution (the topmost branch in Figure 1). As shown in Table 1, model without multi-scale architecture shows significantly worse performance on Accuracy1. The Accuracy1 decreases by 3.1% and 10.9% for *GTzan* and *ACM Mirum* respectively. For Accuracy2, there is also a certain degree of performance degradation. The results demonstrate that the multi-scale can improve the classification ability as well as robustness.

### 3.2 Comparison with Previous Work

To compare with previous work, we use the same test datasets as in [19] (see [14] for details): *ACM Mirum* [9] (1,410 items), *Hainsworth* [32] (222 items), *GTzan* [31] (999 items), *SMC* [33] (217 items), *GiantSteps* [34] (664

| Dataset | böck | schr | foro | mgan |
|---|---|---|---|---|
| ACM Mirum | 74.0 | 79.5 | 73.3 | **82.1** |
| Hainsworth | **80.6*** | 77.0 | 73.4 | 77.5 |
| GTzan | 69.7 | 69.4 | 69.7 | **78.9** |
| SMC | **44.7*** | 33.6 | 30.9 | 29.0 |
| GiantSteps | 58.9 | 73.0 | 83.6 | **90.2** |
| Ballroom | 84.0* | 92.0 | 92.6 | **95.1** |
| ISMIR04 | 55.0 | 60.6 | 61.2 | **61.7** |
| Combined | 69.5 | 74.2 | 74.4 | **79.8** |

(a) Accuracy1

| Dataset | böck | schr | foro | mgan |
|---|---|---|---|---|
| ACM Mirum | **97.7** | 97.4 | 96.5 | 95.7 |
| Hainsworth | **89.2*** | 84.2 | 82.9 | 87.8 |
| GTzan | **95.0** | 92.6 | 89.1 | 91.3 |
| SMC | **67.3*** | 50.2 | 50.7 | 44.7 |
| GiantSteps | 86.4 | 89.3 | **97.9** | 97.6 |
| Ballroom | **98.7*** | 98.4 | **98.7** | 97.7 |
| ISMIR04 | **95.0** | 92.2 | 87.1 | 88.8 |
| Combined | **93.6** | 92.1 | 92.0 | 91.9 |

(b) Accuracy2

**Table 2**: Comparison with the results published by Böck (böck) [15], Schreiber (schr) [19], and Foroughmand (foro) [20]. Best results per test dataset are set in **bold**. Asterisk (*) denotes that the corresponding dataset were used for training.

items), *Ballroom* [3] (698 items), and *ISMIR04* [3] (465 items). The union of all test datasets is referred to as *Combined*. The most recent annotations available are used.

We compare our work (mgan) with previous studies by Schreiber (schr) [19] and Foroughmand (foro) [20]. These two methods are both CNN-based single-step models that we are committed to improve. We consider them as the state-of-the-art among single-step approaches. In addition, we also compare the model with an RNN-based traditional periodicity analysis approach by Böck (böck) [15]. The results are shown in Table 2. Note that *Ballroom*, *Hainsworth*, and *SMC* are used for training in böck (values marked with asterisks *).

Focusing on Accuracy1, the experimental results show that the proposed model surpasses other methods in most cases, which proves the effectiveness of the proposed idea to improve Accuracy1. Especially for *GaintSteps* (664 electronic dance music excerpts), there shows a significant improvement of over 6.6%. The richness of electronic dance music in training data can be considered as a reason. The good performance in *ACM Mirum* and *GTzan* (both multi-genre datasets) shows the generalization potential of our model. Moreover, for *Hainsworth*, the model achieves the highest Accuracy1 among single-step approaches. Finally, the proposed method also reaches the highest Accuracy1 for *Combined* (79.8%) compared with other methods, gaining improvement of 5.4%.

As for Accuracy2, it can be observed that böck achieves the highest accuracy in most cases. Ignoring böck, the proposed model shows a similar performance to other single-step methods.

Among all datasets, the worst results of our model are obtained for *SMC*. The dataset was designed to be difficult to estimate tempo, covering various genres. Although we have tried to supplement and augment the training data, the genre-imbalance problem has not been solved very well. This indicates the necessity to supplement more data with different genres in the future work.

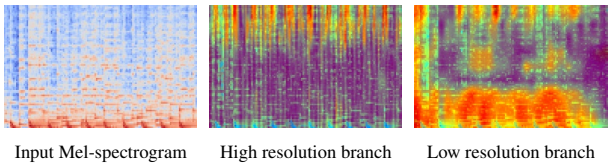
### 3.3 Comparison with Multi-task Approaches

In recent years, some works [17, 18] have not only focused on a single rhythm attribute, but combined the estimation of interconnected rhythm attributes (such as beats, downbeats, etc.) by multi-task learning, so that these highly related tasks can reinforce each other. These approaches are capable of embedding more musical knowledge into a single model, and enrich the training data of each task. In order to further explore the potential of the proposed MGANet and compare its performance with multi-task approaches, we conduct experiments with reference to [17], combining the beat tracking task to our model.

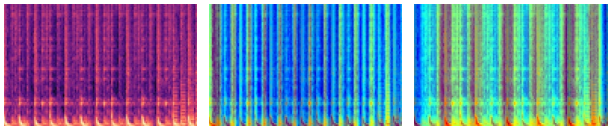
| | Accuracy1 | Accuracy2 |
|---|---|---|
| | *ACM Mirum* | |
| böck19 [17] | 0.749 | 0.974 |
| böck20 [18] | 0.841 | **0.990** |
| mgan | 0.821 | 0.957 |
| mgan+ | **0.846** | 0.970 |
| | *GiantSteps* | |
| böck19 [17] | 0.764 | 0.958 |
| böck20 [18] | 0.870 | 0.965 |
| mgan | **0.902** | **0.976** |
| mgan+ | 0.861 | 0.973 |
| | *GTzan* | |
| böck19 [17] | 0.673 | 0.938 |
| böck20 [18] | **0.830** | **0.950** |
| mgan | 0.789 | 0.913 |
| mgan+ | 0.796 | 0.931 |

**Table 3**: Comparison with multi-task approaches. mgan+ is trained by multi-task learning with beat tracking. Best results per test dataset are set in **bold**.

To predict beat positions, we add a branch to the original network structure. The inputs of the branch are the feature maps before sent into tempo classifier, with shapes of (1, 128, 128), (1, 64, 128), and (1, 32, 128). The low resolution feature maps are up-sampled to 128 frames length on time axis by transposed convolution layers. Then, the concatenated feature map with shape (1, 128, 384) is processed by three $1 \times 3$ convolution layers (output channel number are set to 128, 32, and 1 respectively). After a sigmoid operation, the beat activation function is derived. This extended network structure is trained as a multi-output model to combine the two tasks.

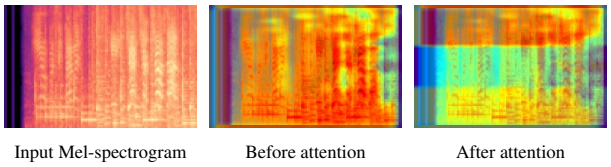



(a) *Samba/Albums-Latin Jam2-14* (0:18-0:24)

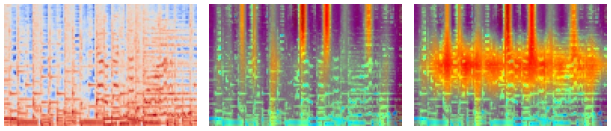



(b) *Chacha/Albums-Latin Jam-01* (0:06-0:12)

**Figure 4**: Grad-CAM visualizations for layers on different resolution branches.




(a) *Chacha/Albums-Latino Latino-0* (0:00-0:06)




(b) *Chacha/Albums-Media-103405* (0:12-0:18)

**Figure 5**: Grad-CAM visualizations for layers before and after grouped attention.

For the training of beat tracking, we use a combination of the following datasets: *Hainsworth* [32], *SMC* [33], *Ballroom* [3], *ISMIR04* [3], *Beatles* [35], and *HJDB* [36]. As for the training of tempo estimation, the training and validation datasets in section 2.2 are used. To further enrich the data, beat annotated datasets are also adopted for the training of tempo classifier, using the average BPMs derived from beat annotations as training labels. We train the two task alternatively every epoch, without changing other experimental settings mentioned in section 2.3.

The experimental results are shown in Table 3. Three datasets *ACM Mirum* [9], *GTzan* [31], and *GiantSteps* [34] are used as test datasets. We compare our works (the original model `mgan` and the multi-task model `mgan+`) with two multi-task approaches `böck19` [17] and `böck20` [18]. By multi-task training, improvement can be observed on *ACM Mirum* and *GTzan*. Especially for *ACM Mirum*, the Accuracy1 is increased by 2.5%, achieving the best result among all approaches. Because the two test datasets are both multi-genre datasets, it can be considered that the good performance comes from not only the multi-task learning, but also the beat tracking datasets with rich music genres. As for *GiantSteps*, `mgan+` performs better than `böck19` and `böck20`, but a bit worse than `mgan`. This is also due to the supplement of data, which affects the dominant position of dance music in training data.

### 3.4 Grad-CAM Analysis

Gradient-weighted Class Activation Mapping (Grad-CAM) [37] is a method that can faithfully highlight the important regions in inputs for a CNN-based classification model. It uses the gradient information in back-propagation as weights (grad-weights) to explain the network's decisions. We visualize the activation maps derived by Grad-CAM as shown in Figure 4 and Figure 5. Red indicates the part more important in predicting tempo while blue contributes less.

Figure 4 shows the activation maps on branches with different resolutions. Their inputs are two audio clips from *Ballroom* dataset. Time duration is marked below the corresponding images, following the audio title set in *italic*. Figure 4a comes from a piece of Samba mainly played by piano and kick drum. The piano in the clip has a higher pitch, played with quarter notes while the kick drum falls on every beat in the bar. It can be observed from the activation maps that the model mainly focuses on short-duration parts of piano in the high-resolution branch, and the kick drum parts with long duration in the low-resolution branch. As for the second example, which is a Cha Cha song, the beat positions can be identified from kick drum in low-frequency part, vocal in middle-frequency part, and claves in high-frequency part. Figure 4b shows that the low-resolution branch considers downbeats to be important, while the high-resolution branch focus on not only downbeats but every other beat in a bar. It can be proved that the multi-scale structure is capable of integrating useful information with different granularities.

We also visualize the activation maps before and after the proposed grouped channel attention to explore the its effect. The results are shown in Figure 5. The music excerpt of Figure 5a is played with regular claves and double bass, hence the high-frequency part and the low-frequency part contribute more to tempo estimation. The attention branch reweights the feature maps from the trunk branch, giving top and bottom parts higher weights to detect tempo information easier. In contrast, the vocal dominates the rhythm information in the song of Figure 5b, thus the model gives higher attention to the middle-frequency part after grouped attention. By grouped attention, the network can efficiently find which part would be considered to be important for tempo estimation.

## 4. CONCLUSION

In this paper, we propose a new CNN-based single-step approach for tempo estimation. We introduce the idea of multi-scale network to construct the architecture of the proposed MGANet. The GAModule with the grouped channel attention is designed to be the key component of the network. Compared with previous work, the proposed approach exhibits good performance on Accuracy1 and outperforms existing models in most cases.

## 5. ACKNOWLEDGEMENT

This work was supported by National Key R&D Program of China (2019YFC1711800), NSFC (61671156).

## 6. REFERENCES

[1] M. Goto and Y. Muraoka, "A beat tracking system for acoustic signals of music," in *Proc. of 2nd ACM International Conference on Multimedia*, 1994, pp. 365–372.

[2] E. D. Scheirer, "Tempo and beat analysis of acoustic musical signals," *The Journal of the Acoustical Society of America*, vol. 103, no. 1, pp. 588–601, 1998.

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

[4] H. Schreiber, "Data-driven approaches for tempo and key estimation of music recordings," Ph.D. dissertation, Friedrich-Alexander-Universität ErlangenNürnberg (FAU), 2020.

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

[6] M. Alonso, G. Richard, and B. David, "Accurate tempo estimation based on harmonic + noise decomposition," *EURASIP Journal on Advances in Signal Processing*, vol. 2007, pp. 1–14, 2007.

[7] A. P. Klapuri, A. J. Eronen, and J. T. Astola, "Analysis of the meter of acoustic musical signals," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 1, pp. 342–355, 2005.

[8] A. T. Cemgil, B. Kappen, P. Desain, and H. Honing, "On tempo tracking: Tempogram representation and kalman filtering," *Journal of New Music Research*, vol. 29, no. 4, pp. 259–273, 2000.

[9] G. Peeters and J. Flocon-Cholet, "Perceptual tempo estimation using gmm-regression," in *Proc. of 2nd International ACM Workshop on Music Information Retrieval with user-centered and multimodal strategies*, 2012, pp. 45–50.

[10] A. Gkiokas, V. Katsouros, and G. Carayannis, "Reducing tempo octave errors by periodicity vector coding and svm learning." in *Proc. of the 13th Int. Society for Music Information Retrieval Conf.*, 2012, pp. 301–306.

[11] G. Percival and G. Tzanetakis, "Streamlined tempo estimation based on autocorrelation and cross-correlation with pulses," *IEEE/ACM Transactions on Audio, Speech, and Language*, vol. 22, no. 12, pp. 1765–1776, 2014.

[12] F.-H. F. Wu and J.-S. R. Jang, "A supervised learning method for tempo estimation of musical audio," in *Proc. of the 22nd Mediterranean Conference on Control and Automation*. IEEE, 2014, pp. 599–604.

[13] F.-H. F. Wu, "Musical tempo octave error reducing based on the statistics of tempogram," in *Proc. of the 23rd Mediterranean Conference on Control and Automation*. IEEE, 2015, pp. 993–998.

[14] H. Schreiber and M. Müller, "A post-processing procedure for improving music tempo estimates using supervised learning," in *Proc. of the 18th Int. Society for Music Information Retrieval Conf.*, 2017, pp. 235–242.

[15] S. Böck, F. Krebs, and G. Widmer, "Accurate tempo estimation based on recurrent neural networks and resonating comb filters," in *Proc. of the 16th Int. Society for Music Information Retrieval Conf.*, 2015, pp. 625–631.

[16] A. Gkiokas and V. Katsouros, "Convolutional neural networks for real-time beat tracking: A dancing robot application," in *Proc. of the 18th Int. Society for Music Information Retrieval Conf.*, 2017, pp. 286–293.

[17] S. Böck, M. E. Davies, and P. Knees, "Multi-task learning of tempo and beat: Learning one to improve the other," in *Proc. of the 20th Int. Society for Music Information Retrieval Conf.*, 2019, pp. 486–493.

[18] S. Böck and M. E. Davies, "Deconstruct, analyse, reconstruct: How to improve tempo, beat, and downbeat estimation," in *Proc. of the 20th Int. Society for Music Information Retrieval Conf.*, 2020, pp. 574–582.

[19] H. Schreiber and M. Müller, "A single-step approach to musical tempo estimation using a convolutional neural network," in *Proc. of the 19th Int. Society for Music Information Retrieval Conf.*, 2018, pp. 98–105.

[20] H. Foroughmand and G. Peeters, "Deep-rhythm for tempo estimation and rhythm pattern recognition," in *Proc. of the 20th Int. Society for Music Information Retrieval Conf.*, 2019, pp. 636–643.

[21] D. Gärtner, "Tempo detection of urban music using tatum grid non negative matrix factorization," in *Proc. of the 14th Int. Society for Music Information Retrieval Conf.*, 2013, pp. 311–316.

[22] G. Huang and D. Chen, "Multi-scale dense networks for resource efficient image classification," in *Proc. of the International Conference on Learning Representations*, 2018.

[23] J. Wang, K. Sun, T. Cheng, B. Jiang, C. Deng, Y. Zhao, D. Liu, Y. Mu, M. Tan, X. Wang *et al.*, "Deep high-resolution representation learning for visual recognition," *IEEE transactions on pattern analysis and machine intelligence*, 2020.

[24] A. A. Adegun and S. Viriri, "Fcn-based densenet framework for automated detection and classification of skin lesions in dermoscopy images," *IEEE Access*, vol. 8, pp. 150 377–150 396, 2020.

[25] B. Xiao, H. Wu, and Y. Wei, "Simple baselines for human pose estimation and tracking," in *Proc. of the European Conference on Computer Vision*, 2018, pp. 466–481.

[26] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and accurate deep network learning by exponential linear units (elus)," *arXiv preprint arXiv:1511.07289*, 2015.

[27] Y. Cao, J. Xu, S. Lin, F. Wei, and H. Hu, "Gcnet: Non-local networks meet squeeze-excitation networks and beyond," in *Proc. of the IEEE/CVF International Conference on Computer Vision Workshops*, Oct 2019.

[28] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7132–7141.

[29] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka, "Rwc music database: Popular, classical and jazz music databases," in *Proc. of the 3rd Int. Society for Music Information Retrieval Conf.*, vol. 2, 2002, pp. 287–288.

[30] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. of the 3rd International Conference on Learning Representations*, 2015.

[31] U. Marchand, Q. Fresnel, and G. Peeters, "Gtzan-rhythm: Extending the gtzan test-set with beat, downbeat and swing annotations," in *Extended abstracts for the Late-Breaking Demo Session of the 16th International Society for Music Information Retrieval Conf.*, 2015.

[32] S. W. Hainsworth, "Techniques for the automated analysis of musical audio," Ph.D. dissertation, University of Cambridge, 2004.

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

[34] P. Knees, Á. Faraldo Pérez, H. Boyer, R. Vogl, S. Böck, F. Hörschläger, M. Le Goff *et al.*, "Two data sets for tempo estimation and key detection in electronic dance music annotated from user corrections," in *Proc. of the 16th Int. Society for Music Information Retrieval Conf.*, 2015, pp. 364–370.

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

[36] J. Hockman, M. E. Davies, and I. Fujinaga, "One in the jungle: Downbeat detection in hardcore, jungle, and drum and bass." in *Proc. of the 13th Int. Society for Music Information Retrieval Conf.*, 2012, pp. 169–174.

[37] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *Proc. of the IEEE International Conference on Computer Vision*, 2017, pp. 618–626.