THE 2025 KG MUSIC BEATS TRACKING SYSTEM

DingKun Xiao
KG Music Apollo Lab
stefanxiao@kugou.net

Haijun Cai
KG Music Apollo Lab
andrewcai@kugou.net

Chuanyi Chen
KG Music Apollo Lab
chenchuanyi@kugou.net

ABSTRACT

This paper describes the ApolloBeats, our submission to the MIREX 2025 Audio Beat Tracking Task[1]. We propose a method that addresses the limitations of existing datasets and models by enhancing high-quality, and diverse data and specific model architectures. Our approach utilizes a meticulously annotated MbeatV1 dataset, incorporating multiple genres and mixed meters to capture the diversity of musical audio, and the support for meter changes. We adopt BeatThis[2] as the baseline model, modifying its modules and introducing multi-task prediction to improve beat-tracking accuracy. We demonstrate that combining a rich, high-quality dataset with a customized model architecture significantly enhances the accuracy of musical beat detection, particularly in handling complex scenarios involving mixed and changing meters.

1. INTRODUCTION

Beat tracking is the task of estimating the temporal locations of musical beats in an audio signal. It is often combined with the downbeat tracking task, which targets a higher metrical level: tracking the beginning of each measure. However, despite its seemingly straightforward nature, practical implementation faces numerous technical challenges, primarily stemming from the inherent complexity of music and the difficulties in signal processing. For instance, musical pieces of different styles and rhythms may exhibit similar time-domain characteristics, making it difficult for models to accurately identify beat positions—especially in scenarios involving significant noise, overlapping instruments, or rhythmic variations. Additionally, traditional feature extraction methods often show limitations when dealing with non-stationary audio signals, further complicating beat detection.

To address these challenges, we introduce ApolloBeats, a model based on Transformer architecture and high-quality data, which enhances temporal modeling and context-aware mechanisms, significantly improving the robustness and generalization of beat tracking.

2. METHOD

2.1 Models

Our model is based on the modifications made to BeatThis[2], and it has been restructured into three modules:

1. **Audio Encoder**: Uses the PartialF-T Encoder. Later, we will consider using the Whisper Encoder and ViT Encoder commonly used in MLLM;
2. **MLP Adapter**: Feature mapping layer, which maps the input of the audio encoder into audio tokens;
3. **Transformer**: Temporal prediction module, which predicts the meter corresponding to the audio tokens.

In addition, we have also added two prediction branches for other tasks: bpm (the unit of beats per minute) prediction and meter (a symbol marked in fraction form in the score, used to represent the rhythm structure of the music piece,, such as 2/4, 4/4, etc.) classification. That is, we added two class tokens to the above Transformer to predict bpm and meter. Experiments have proved that this multi-tasking can improve the robustness and accuracy of the beat tracking detection task.

2.2 Dataset

Our training data consists of two parts: one is an open-source dataset, and the other is our own annotated MbeatV1 dataset:

1. The open-source dataset[3]: Simac, SMC, HJDB, Beatles, Harmonix, RWC (classical, pop, royalty-free, and jazz), TapCorrect, JAAH, Filosax, ASAP, Groove MIDI, GuitarSet, Candombe. Gtzan, etc.
2. The MbeatV1 dataset: Since 80% of the samples in the open-source dataset are audio clips of 20-30 seconds and the audio songs are relatively old, this is not friendly for our long-time sequence modeling and long-context-aware ApolloBeats as well as modern song beat prediction. Therefore, we have constructed a brand-new and diverse MbeatV1 dataset of approximately 5,000 songs, covering various genres, including pop, rock, folk, various sub-genres of electronic dance music, etc. This dataset covers multiple regions, including China, Japan, South Korea, and Europe and America.

2.3 Training

Firstly, we trained 200 epochs on the open-source dataset using audio clips of 30 seconds each. Then, we fine-tuned on our MbeatV1 dataset with clips of 1 minute each for another 200 epochs. During training, every time a sample is drawn, we randomly select a data augmentation. The full training takes around 20 hours on a 8-GPU machine NVIDIA L20 with the mixed precision of bf16.

3. REFERENCES

[1] https://www.music-ir.org/mirex/wiki/2025:Audio_Beat_Tracking

[2] Foscarin, Francesco , Schlüter, Jan, and G. Widmer . "Beat this! Accurate beat tracking without DBN postprocessing." (2024).

[3] https://zenodo.org/records/13922116