Transactions of the International Society for Music Information Retrieval

XXXX, X., and XXXX, X. (2024). BeatNet+: Real-Time Rhythm Analysis for Diverse Music Audio, *Transactions of the International Society for Music Information Retrieval*, V(N), pp. xx–xx, DOI: https://doi.org/xx.xxxx/xxxx.xx

ARTICLE TYPE

BeatNet+: Real-Time Rhythm Analysis for Diverse Music Audio

XXX XXX,* and XXX XXX

Abstract

This paper presents a comprehensive study on real-time music rhythm analysis, covering joint beat and downbeat tracking for diverse kinds of music signals. We introduce BeatNet+, a two-stage approach to real-time rhythm analysis built on a previous state-of-the-art method named BeatNet. The main innovation of the proposed method is the auxiliary training strategy that helps the neural network model to learn a representation invariant to the amount of percussive components in the music. Together with other architectural improvements, this strategy significantly improves the model performance for generic music. Another innovation is on the adaptation strategies that help develop real-time rhythm analysis models for challenging music scenarios including isolated singing voices and non-percussive music. Two adaptation strategies are proposed and experimented with different neural architectures and training schemes. Comprehensive experiments and comparisons with multiple baselines are conducted, and results show that BeatNet+ achieves superior beat tracking and downbeat tracking F1 scores for generic music, isolated singing voices and non-percussive audio, with competitive latency and computational complexity. Finally, we release beat and downbeat annotations for two datasets that are designed for other tasks, and revised annotations of three existing datasets. We also plan to release the code repository and pre-trained models on GitHub.

Keywords: Real-time beat tracking, downbeat tracking, rhythm analysis, singing voices, non-percussive music, BeatNet, BeatNet+

1. Introduction

Music can be regarded as one of the most intricate and diverse art forms in the world. It is created through the incorporation of various sounds that are arranged in a meaningful manner to produce a unique composition. One of the key elements of music is rhythm, which refers to the sequential pattern of sounds and silences that occur over time. Rhythm is crucial in music as it forms the fundamental basis upon which a piece is constructed. In recent years, there has been an increasing interest in developing real-time music rhythm analysis systems Heydari and Duan (2021).

Accurate and robust real-time music rhythm analysis holds the potential to advance the music industry, enabling innovative applications. It can serve as a fundamental component for a variety of use cases, including automatic music generation, processing and analysis. With the recent advancements in virtual and augmented reality, there is a growing demand for real-time music processing and analysis across various situations. This need has also gained prominence due to its role in empowering the creation of immersive music-based interactive experiences. These experiences, include but are not limited to real-time music visualization (Bain, 2008), dancing robots (Bi et al., 2018), DJing and live remixing and sampling performance (Cliff, 2000), live video editing and synchronization (Davis and Agrawala, 2018), dynamic lighting systems, and music-driven interactive video games (Bégel et al., 2018), offer users the chance to engage with music on the fly.

Developing real-time music rhythm analysis systems involves addressing three key challenges: The first challenge is on maintaining high accuracy while not accessing future input data as offline models do. The second challenge is on achieving low latency, especially on low-powered devices. While the first two challenges are easy to understand, we argue that the third challenge is on the generalization to various kinds of music audio. While state-of-the-art rhythm analysis

XXXX, XXXX, XXX,XXX

research has shown promising performance on music recordings that contain strong percussive components (e.g., drums, rhythmic guitar, piano) (Heydari et al., 2021), there are scenarios where the music audio lacks such components. For example, real-time rhythm analysis of isolated singing voices plays a crucial role in understanding and processing vocal performances and it enables applications such as accompaniment generation on the fly and live music remixing (Heydari et al., 2023). As another example, real-time generation of drum tracks requires rhythm analysis of non-percussive music tracks and can enable collaborative music making between human musicians and artificial intelligence (AI) agents.

In this work, we propose BeatNet+ for real-time rhythm analysis for diverse kinds of music audio. Similar to BeatNet (Heydari et al., 2021), BeatNet+ processes the music audio magnitude spectrum with a Convolutional Recurrent Neural Network (CRNN) to compute beat and downbeat activations in each audio frame. The activations are then post-processed by a two-level cascade Monte Carlo particle filter. The key innovations of BeatNet+ are on an auxiliary training strategy that improves the system performance over state-of-the-art rhythm tracking methods on generic music, as well as adaptation strategies that improve the generalization ability to less percussive music such as isolated singing voices and music without drums, which are novel rhythm analysis settings.

Specifically, the auxiliary training strategy leverages a parallel regularization branch that has an identical structure without weight sharing with the main branch (i.e., used for inference) during training. The main branch is fed with full music mixtures while the auxiliary branch is fed with the full music less drum tracks (referred to as *non-percussive* versions) of the same pieces. In addition to the Cross Entropy (CE) losses for each branch, a Mean Squared Error (MSE) loss is computed between the latent embeddings of the two branches to regularize the representation learning of the main branch.

Regarding the adaptation strategies for BeatNet+ to work with less percussive music, we propose two techniques termed *Auxiliary-Freezing (AF)* and *Guided Fine-tuning (GF)*. The AF approach (Figure 2) again adopts a two-branch auxiliary training strategy similar to the one mentioned above. Differently, the main branch (left) is now trained on the target music type (i.e., less percussive music) while the auxiliary branch (right) is frozen as the pre-trained main branch of the BeatNet+ which is trained for full music mixtures. The GF technique (Figure 3) employs a single-branch model initialized with the pre-trained main branch of the BeatNet+ model. Subsequently, this model undergoes fine-tuning on input music pieces, starting with full music mixtures (aligned with the original data type of the pre-trained model), which are gradually adapted to match the target music type. For instance, if the target is isolated singing voices, non-singing parts of the music input are progressively removed during training iterations. We perform experiments on two types of less percussive music types to demonstrate the effectiveness of the adaptation strategies: *Isolated singing voices* and *non-percussive music*. Rhythm tracking for both settings is novel and could enable novel applications such as real-time drum track generation.

Finally, we release beat and downbeat annotations of MUSDB18 (Rafii et al., 2017) and URSing (Li et al., 2021) datasets, which were originally designed for other MIR tasks, enabling them to be utilized for music rhythm analysis applications. Also, we correct mistakes in the rhythm annotations of three pre-existing music rhythm analysis datasets including RWC jazz, RWC pop, and RWC royalty-free (Goto et al., 2002; Goto, 2004). The source code of the BeatNet+, adaptation models and rhythmic annotations of MUSDB and URSing will be online[1].

2. Related Work

Existing work on rhythm analysis can be reviewed along different dimensions. In this section, we provide a review along the dimensions that are related to the proposed work.

2.1 Two-Stage Approach

The majority of rhythm analysis methods (e.g., beat tracking, downbeat tracking) adopt a two-stage approach. In the first stage, a salience function (also called likelihood function, detection function, or activation strength) is computed from the input audio signal to represent the salience of the target event (e.g., a beat) in different time frames. In the second stage, an inference process (also called post-processing) is employed to make binary decisions on the presence of the target event in each audio frame based on the salience function. Different techniques have been proposed in each of these stages, and we will review them in the following.

2.1.1 Salience Calculation Stage

There are generally two paradigms in computing the salience function. The first paradigm is rule-based and uses hand-crafted functions to indicate the presence of important rhythmic elements in music, such as onsets and beats (Mottaghi et al., 2017; Chiu et al., 2023). Such function often describes the “novelty” of the current audio frame compared to the previous frame(s) in terms of energy (Schloss, 1985) and spectral content (Masri, 1996). These hand-crafted functions are generally fast to compute and robust to music styles. However, their detection accuracy is limited compared to data-driven methods in the next paragraph.

[1] [We open-source the following upon the paper acceptance:]
Codes: https://github.com/XXXXX/XXXXXX
Annotations: https://github.com/XXXXX/XXXXXX

The second paradigm focuses on machine learning techniques, where models are trained to establish the relationship between low-level acoustic features and annotations of rhythmic elements (Holzapfel et al., 2012; Gkiokas et al., 2012; Böck and Schedl, 2011; Böck et al., 2016). Deep learning-based methods have gained significant attention due to their exceptional performance in rhythm analysis. These models typically require supervision and are trained on large datasets with labeled rhythmic patterns, making them highly accurate in recognizing complex rhythmic patterns. They leverage neural networks to extract "activation strength" for every time frame. Several neural network structures are utilized for music rhythm analysis tasks such as convolutional networks (Gkiokas and Katsouros, 2017), cepstroid invariant networks (Elowsson, 2016), recurrent networks (Eyben et al., 2013), transformers (Heydari and Duan, 2022), temporal convolutional networks (Davies and Böck, 2019), and autoencoders (Greenlees, 2020).

Recently, self-supervised learning (SSL) models have gained popularity as they can be trained on massive amounts of unlabeled data. Desblancs et al. (2023) proposed ZeroNS that leverages a self-supervised pre-processing block for their beat tracking model. Similar to our proposed BeatNet+ model, ZeroNS contains two branches and leverages different music stems in training. However, there are several fundamental differences between the two models. BeatNet+ is a supervised model with a latent matching loss, whereas ZeroNS is self-supervised and lacks a loss-matching regularization term. BeatNet+ focuses on the causal joint beat and downbeat tracking, while ZeroNS serves as a non-causal model designed only for beat tracking. In terms of structure, BeatNet+ utilizes CRNN networks, while ZeroNS only incorporates convolutional blocks in its pipeline. SSL representations have also been used in rhythm analysis of challenging music inputs such as isolated singing voice (Heydari and Duan, 2022). Such representations, however, can be difficult to use in real-time applications due to causal and low latency requirements.

It is worth mentioning that each of the mentioned methods can operate in either the time domain, e.g., (Steinmetz and Reiss, 2021; Heydari and Duan, 2022) or frequency domain, e.g., (Meier et al., 2021; Böck and Davies, 2020; Chiu et al., 2023), or combined, e.g., (Morais et al., 2023). Time-domain techniques operate on the audio waveform, while frequency-domain techniques operate on a time-frequency representation computed from Fourier, constant-Q or other transforms. They provide explicit information about the signal's frequency components and are known for their robustness to noise when compared with time-domain techniques (Zhengqing and Jian-hua, 2005). Spectral approaches face a time-frequency resolution trade-off where extending the time window captures lower frequencies beneficial for rhythm analysis but reduces time resolution, and vice versa. To tackle the time-frequency resolution tradeoff issue, some works, e.g., (Böck et al., 2014), employ multi-resolution embeddings, which involve concatenating spectral features calculated based on different window lengths.

2.1.2 *Decision Stage*

Depending on whether future audio frames are considered in making the prediction at the current frame, the decision stage can be categorized as *offline* and *online* methods. Offline methods, such as comb filters (Scheirer, 1998), dynamic programming (Ellis, 2007), and dynamic Bayesian networks (Böck et al., 2014), improve prediction coherence but are unsuitable for real-time use. A sliding window framework allows offline methods to work in online scenarios (Davies et al., 2005), processing only signals within the window. However, this ignores past signals outside the window, affecting coherence. Overlapping windows can cause computational overload as well.

In online (especially real-time) scenarios, various inherently causal inference methods are utilized, including the forward algorithm (Federgruen and Tzur, 1991), Kalman filtering (Shiu and Kuo, 2007), particle filtering (Hainsworth and Macleod, 2004; Heydari and Duan, 2021; Heydari et al., 2023) and jump-reward inference (Heydari et al., 2022). In particular, particle filtering uses particles to represent and evolve the posterior distribution of rhythmic states like beat, downbeat, and non-beat over time. Hainsworth and Macleod (2004) applied it to tempo detection and Heydari et al. (2021) applied it to joint beat, downbeat, and time signature tracking.

Particle filtering faces challenges in capturing extended temporal dependencies like time signature tracking due to its Markovian nature, relying only on current state predictions. Heydari et al. (2023) proposed dynamic particle filtering, enhancing inference by incorporating historical and salience information, albeit with increased computational cost. Particle filtering also requires numerous particles for extensive state spaces, crucial for detailed time granularity and broad tempo ranges in rhythm analysis, leading to higher computational overhead. Heydari et al. (2022) introduced "jump-reward inference," a semi-Markovian model operating in a 1-dimensional state space, significantly cutting computation time, albeit with a performance drop in higher-level music analysis tasks such as downbeat tracking.

2.2 Real-Time Systems

In this subsection, we briefly review a few real-time beat and downbeat tracking systems. IBT (Oliveira et al., 2010) is a signal processing based multi-agent system for real-time beat tracking. It initializes a set of agents with various hypotheses. Each agent carries a hypothesis concerning the rate and placement of musical beats and the model dynamically chooses the best agent based on music onsets.

In the realm of deep learning based methods, Böck et al. (2014) employed an RNN to compute activations and apply the forward algorithm (Federgruen and Tzur, 1991) for inferring beats in a causal setting. Heydari et al. (2021) proposed BeatNet, a real-time system for joint beat, downbeat, and meter tracking. It employs a fully causal CRNN structure with a 1D convolutional layer to produce three activations for beat, downbeat, and non-beat. It uses an efficient two-level particle filtering for inference. In their follow-up work, Heydari et al. (2022) utilized BeatNet activations and presented a so-called "jump-back reward" strategy to speed up the particle filtering process as reviewed in the previous subsection.

Chang and Su (2024) proposed an online beat and downbeat tracking system named BEAST based on the streaming Transformer Tsunoo et al. (2019). Through the incorporation of contextual block processing in the Transformer encoder and relative positional encoding in the attention layer, BEAST achieves significant improvements over existing state-of-the-art models. It uses the forward algorithm (Federgruen and Tzur, 1991) as the inference stage.

2.3 Rhythm Analysis for Isolated Singing Voices

In order to address the isolated singing voice rhythm analysis task, Heydari and Duan (2022) proposed a model that leverages pre-trained self-supervised speech models such as WavLM (Chen et al., 2022) and Distilhubert (Chang et al., 2022) and built some linear transformers (Katharopoulos et al., 2020) on top of them to jointly extract the beats of singing voices in an offline fashion. This study highlights the substantial performance improvement achieved by utilizing pre-trained speech models and transformers. Nonetheless, their computational heaviness poses challenges for real-time and low-resource applications, especially in scenarios with limited computational power, such as in-device use cases. SingNet (Heydari et al., 2023) pioneered real-time singing voice joint beat and downbeat, and meter tracking. It utilizes a slightly larger CRNN model compared to BeatNet for calculating activation functions. Recognizing the irregular and noisy activations delivered by singing voices, SingNet introduces dynamic particle filtering, a novel inference module that incorporates offline estimation and activation saliences into the online inference process.

2.4 Rhythm Analysis for Non-Percussive Music

In addition to isolated singing voices, there are other types of music audio that are less percussive, e.g., music without drums. Real-time music rhythm analysis for these kinds of music is also challenging but can be very useful in many applications such as the automatic generation of drum tracks. Wu et al. (2022) developed an offline drum accompaniment system based on an offline drum-aware beat tracking method (Chiu et al., 2021). Online rhythm analysis of non-percussive music, however, is limited to a few traditional signal processing approaches such as (Goto, 2001; Goto and Muraoka, 1999) that only track beats but not downbeats or meter.

3. Methodology

In this section, we present a novel two-stage approach named BeatNet+ to real-time joint beat, downbeat and meter tracking for diverse kinds of music inputs. The first stage estimates beat and downbeat saliences from audio frames, while the second stage makes decisions using particle filtering. Additionally, we elaborate on adapting the BeatNet+ model for rhythm analysis of more challenging data types.

3.1 Stage 1: Beat and Downbeat Salience Estimation

This section describes the proposed neural network model and training strategies for robust computation of beat and downbeat saliences from diverse kinds of music inputs.

3.1.1 Audio Feature Representation

We utilize Short-Time Fourier Transform (STFT) to compute a log-magnitude spectrogram as the input feature representation. The window length is set to 80 ms with a Hann window. The window hop size, i.e., the model's theoretical latency, is set to 20 ms. The frequency range is between 30 Hz and 17,000 Hz with 288 bins.

3.1.2 Neural Architecture and Training Strategy

BeatNet+ (Figure 1) features two branches where both the main branch (left) and the auxiliary branch (right) are used in training while for inference, only the main branch is utilized. Both branches employ a convolutional recurrent neural network (CRNN) structure similar to BeatNet (Heydari et al., 2021), where the convolutional block is identical to that of BeatNet but the recurrent block is expanded from two layers to four layers based on preliminary empirical studies. This deeper design is reasonable, as BeatNet+ is expected to handle diverse music inputs, including isolated singing voices and less-percussive music with complex rhythmic structures. Each recurrent layer contains 150 long short-term memory (LSTM) cells, the same as in BeatNet. It is worth mentioning that in our pilot study, we explored various alterations to the neural architecture, such as incorporating batch normalization, linear layers, Rectified Linear Unit (ReLU) activations, and leaky ReLU activations. However, these modifications did not yield significant performance improvements.

To increase the robustness to music with various levels of percussive components, we use an auxiliary branch (the right branch of Figure 1) to train Beat-

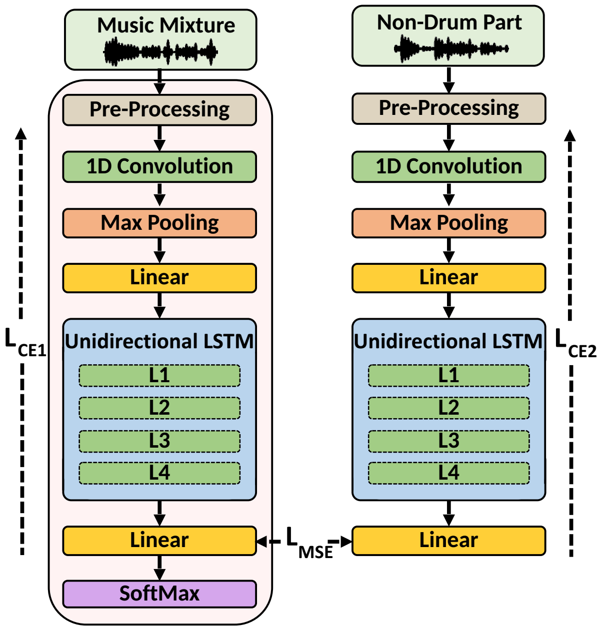

Figure 1: Neural structure of *BeatNet+* for general music rhythm analysis. Both the main (left) and auxiliary (right) branches are initialized randomly and trained jointly, but only the main branch is utilized for inference.

Net+. The auxiliary branch is identical to the main branch, except that it takes a different type of input during training and it does not include the SoftMax layer, which is only used during inference in the main branch. Note that since cross-entropy loss with logits is being used, applying SoftMax is unnecessary during training.

Training of BeatNet+ takes three losses as in Equation (1):

$$L_{total} = L_{CE1} + L_{CE2} + \lambda L_{MSE}. \tag{1}$$

The main branch is trained on full music mixtures with a cross-entropy loss denoted as L_{CE1}. The auxiliary branch is trained on the non-percussive parts of the same music mixtures with another cross-entropy loss denoted as L_{CE2}. Additionally, we introduce a Mean Squared Error (MSE) loss, L_{MSE}, between intermediate representations of the two branches. This can be viewed as a training regularization to encourage similarity between the latent representations of the two branches, given that their outputs, i.e., their rhythm information, are expected to be identical. Based on our pilot studies, Mean Squared Error (MSE) is found to be more suitable than other losses like Mean Absolute Error (MAE) or Huber loss for this regularization. The constant weight parameter λ controls the strength of the regularization. A similar latent matching strategy has been used before to enhance a talking face generation model's robustness to noise (Eskimez et al., 2019).

3.1.3 *Adaptation for More Challenging Music Inputs*

To address the real-time rhythm analysis of challenging inputs such as isolated singing voices and other less-percussive music, we propose two adaptation strategies named as *Auxiliary Freezing (AF)* and *Guided Fine-tuning (GF)*, respectively. Here we take the isolated singing voice scenario as an example, but the proposed adaptation strategies can be applied to other scenarios, e.g., non-percussive music, as well. In the AF approach (shown in Figure 2), we adopt a similar two-branch auxiliary training approach to that in Section 3.1.2. In this case, the auxiliary branch (right) is initialized with the frozen weights from the pre-trained main branch of BeatNet+ (i.e., left branch in Figure 1) taking full music mixtures as inputs, while the main branch (left), is trained from scratch on isolated singing voices of the corresponding music mixtures. MSE loss is imposed between the latent representations of the two branches in addition to the cross entropy loss of the right branch. After this adaptation, the main branch (left) is used for rhythm analysis of isolated singing voices. Note that this approach bears similarity to teacher-student model distillation methods e.g., Kim and Rush (2016), wherein the student model is trained to replicate similar latents as the frozen teacher model. However, the key distinction lies in the fact that commonly used teacher-student models try to perform model distilla-

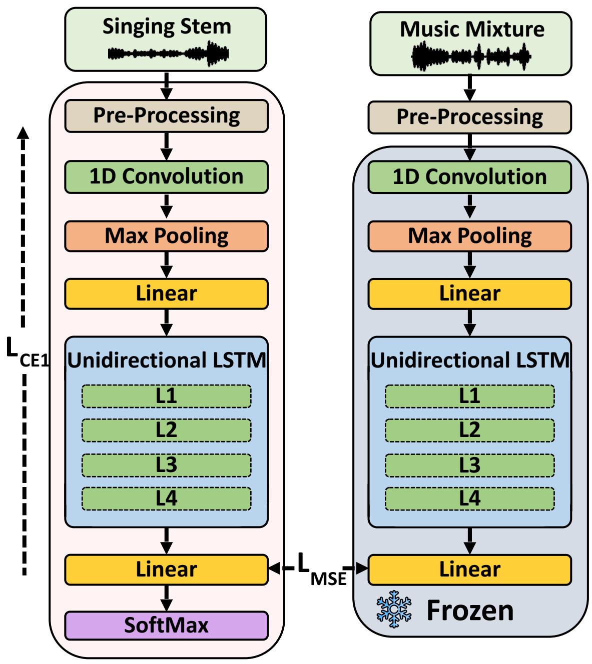

Figure 2: Neural structure of *Auxiliary-Freezing (AF)* adaptation approach for singing voice rhythm analysis. The main branch (left) is initialized randomly and trained for real-time inference, while the auxiliary branch (right) is initialized with the pre-trained BeatNet+ main branch weights and remains frozen during training.

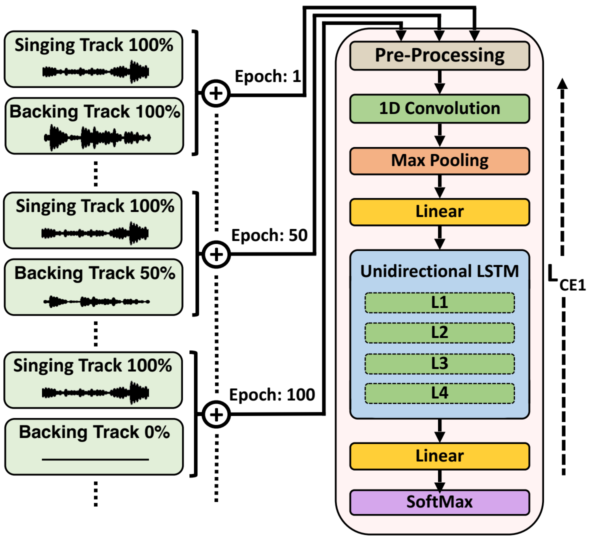

Figure 3: Illustration of the *Guided Fine-tuning (GF)* approach for singing voice rhythm analysis. The model is initialized with the pre-trained BeatNet+ main branch weights and fine-tuned using music mixtures with backing music gradually removed over training epochs.

tion, i.e., to attain similar results with smaller networks on the same data, while our model's objective is to achieve similar results with identical networks on different but related data.

In the Guided Fine-tuning (GF) approach, we commence by initializing a single-branch model with the weights and biases of the main branch of BeatNet+ that is pre-trained on full music mixtures, i.e., the left branch of Figure 1. Subsequently, we fine-tune the model for isolated singing voices by gradually reducing the intensity of the accompanying music during training. In each epoch, a percentage of the accompanying music is deducted, with a linear decay factor denoted as γ. After a number of epochs, the strength of accompanying music in the training data diminishes to zero. Figure 2 illustrates this adaptation approach for isolated singing voice music with $\gamma = 0.01$.

As previously mentioned, both adaptation strategies can be applied to address different types of less-percussive music input. For instance, in Figures 2 and 3, substituting the singing stem with complete musical mixtures excluding drum stems, enables the models to be trained specifically for non-percussive music.

3.2 Stage 2: Decision

Since Cascade Monte Carlo particle filtering demonstrated superior performance for online rhythm analysis tasks among the proposed methods (Heydari et al., 2021, 2023), we use it as the decision-making block for all proposed methods and scenarios. In this section, we provide a brief description of the method we used.

3.2.1 *State Space, Transition and Observation Models*

The state space, transition, and observation models mirror those of BeatNet (Heydari et al., 2021). We implement the discrete 2D state space proposed in (Krebs et al., 2015) and adapt BeatNet's cascade approach. In this approach, instead of merging multiple beat state spaces into a bar state space, two separate state spaces are employed, one for beat and tempo tracking and the other for downbeat and meter tracking, organized hierarchically. The first space comprises tempo and beat phase as the two dimensions; Adjacent states with the same tempo correspond to adjacent time frames of audio. The second space comprises meter (represented as the number of beats per bar) and downbeat phase as the two dimensions; Adjacent states with the same meter correspond to adjacent beats in time. Transition models permit tempo and meter changes to update at beat and downbeat positions, respectively. Observation models calculate beat and downbeat likelihoods based on salience estimated by the neural network.

3.2.2 *Causal Inference*

Monte Carlo particle filtering is a top choice for real-time inference due to two key advantages. Firstly, it does not rely on future data, unlike popular maximum a posteriori (MAP) algorithms such as the Viterbi algorithm and smoothing algorithms like forward-backward. Secondly, unlike many inference algorithms such as Kalman filtering which require strong distribution type assumptions, it is a general and non-parametric approach, capable of decoding any unknown distribution among causal filtering methods. Some previous works (Heydari et al., 2021; Heydari and Duan, 2021) demonstrated its superiority compared to other inference models.

Particle filtering is a two-step inference process that encompasses the *predict/motion* step and the *update/correction* step. In the motion step, particle positions are updated based on predicted trajectories, while the correction step involves adjusting particles and assigning weights based on observed data compatibility. Given the latent state ϕ_k and observation y_k at frame k, assuming that the current position posterior $p(\phi_k|y_{1:k})$ is estimated, the "predict-update" procedure computes the next frame's position posterior $p(\phi_{k+1}|y_{1:k+1})$. Equation (2) details the motion step for one-step-ahead prediction by applying the state transition model $p(\phi_{k+1}|\phi_k)$ into the current frame posterior,

$$p(\phi_{k+1}|y_{1:k}) = \sum_{\phi_k} p(\phi_{k+1}|\phi_k)p(\phi_k|y_{1:k}). \qquad (2)$$

Equation (3) describes the correction step by incorporating the observation likelihood $p(y_{k+1}|\phi_{k+1})$ into the one-step-ahead prediction to estimate the next step posterior,

$$p(\phi_{k+1}|y_{1:k+1}) = \frac{1}{Z_{k+1}} p(y_{k+1}|\phi_{k+1})p(\phi_{k+1}|y_{1:k}). \qquad (3)$$

By combining these motion and correction steps iteratively, particle filtering refines the estimation of the system's state, making it a powerful technique for tracking and inference in dynamic environments.

4. Experiments

In this section, we discuss the training specifics of the proposed models. We also describe the details of our comparison methods, utilized datasets (existing and annotated), and the evaluation metrics for each task. Finally, we report the experimental results for all of the models and compare them with state-of-the-art methods for each task. Note that all experiments with the proposed methods employ the same inference method i.e., the particle filtering approach proposed in BeatNet (Heydari et al., 2021).

4.1 Datasets

To increase data diversity, we use multiple music audio datasets with beat and downbeat annotations, as shown in Table 1. Among these datasets, Ballroom (Gouyon et al., 2006; Krebs et al., 2013), GTZAN (Marchand and Peeters, 2015; Tzanetakis and Cook, 2002), Hainsworth (Hainsworth and Macleod, 2004), Rock Corpus (De Clercq and Temperley, 2011), and RWC Jazz, Pop and Royalty-free datasets (Goto et al., 2002; Goto, 2004) already come with beat and downbeat annotations. However, some downbeat annotations of RWC Jazz, Pop and Royalty-free datasets are not accurate, and we revise them manually. In addition, MUSDB18 (Rafii et al., 2017) and URSing (Li et al., 2021) are multi-track singing datasets without beat or downbeat annotations, and we annotate them using BeatNet (Heydari et al., 2021) followed by manual corrections.

Following the previous works, we employ the whole GTZAN dataset as the test set, given that it is one of the largest and most genre-inclusive datasets for our tasks. Importantly, none of the reported models have been exposed to this dataset during their training phase, ensuring a fair and unbiased assessment. The rest of the datasets outlined in Table 1 are utilized for training and validation purposes.

It is noted that to obtain the audio stems of the datasets for different tasks except the ones that include separate stems i.e., MUSDB18 and URSing, we utilize Demucs (Défossez, 2021), a top-performing open-source music source separation model. It separates each piece of music into four tracks: bass, drums, vocals and others.

For the isolated singing rhythm analysis task, the availability of singing stems is essential. Yet, in the datasets we use, many pieces do not have singing, and some have extended segments with only instrumental music and no vocals. To address this challenge, we introduce a preprocessing stage designed to eliminate vocal-less pieces and extended segments without singing. This is achieved by implementing energy-based vocal Root Mean Square (RMS) thresholding on separated singing tracks. As a consequence, datasets such as RWC-Jazz (Goto et al., 2002; Goto, 2004) were entirely excluded from the data pool for the singing voice rhythm analysis task. Furthermore, some vocal tracks containing extended silent intervals are split into shorter vocal segments.

4.2 Evaluation Metrics

The reported metrics comprise beat and downbeat F1 scores, system latency, and real-time Factor (RTF). Following the literature, F1 scores are reported with a tolerance window of 70 ms. Latency is defined as the hop size of the Short-Time Fourier Transform (STFT) for processed data. RTF is another important metric for real-time models and refers to the speed or responsiveness with which a model can process and generate outputs in real-time. It is the averaged ratio between the total processing time and the total audio length across the whole test set. Note that the reported RTFs are measured on a Windows machine with an AMD Ryzen 9 3900X CPU and 3.80 GHz clock frequency.

Previous work (Heydari et al., 2023) used 200 ms as the tolerance for singing voice beat and downbeat tracking. This was based on their observation that human tolerance to beat and downbeat timing deviations tends to be more lenient for less percussive music compared to music with strong percussions. Therefore, we also report F1 scores with a tolerance of 200 ms for singing voice and non-percussive music datasets in addition to the standard 70 ms tolerance.

4.3 Comparison Methods

To assess the effectiveness of the auxiliary training strategy in Section 3.1.2, we trained two models: **BeatNet+** is the proposed model with auxiliary training using two branches, and **BeatNet+ (Solo)** trains the main branch without the auxiliary branch, i.e., only L_{CE1} is used in Equation (1).

To evaluate BeatNet+ model on real-time rhythm analysis for generic music, we compare it with five baseline models. 1) **BeatNet** (Heydari et al., 2021)

Dataset	*#Pieces*	*#Vocals*	*Labels*
Ballroom	699	452	Original
GTZAN	999	741	Original
Hainsworth	220	154	Original
Rock Corpus	200	315	Original
MUSDB18	150	263	Added
URSing	65	106	Added
RWC jazz	50	0	Revised
RWC pop	100	188	Revised
RWC Royalty-free	15	29	Revised

Table 1: Datasets used in our experiments. GTZAN is used for evaluation and the others are used for training and validation.

employs a CRNN structure and proposes efficient particle filtering for joint beat, downbeat, and meter tracking. 2) **Novel 1D** (Heydari et al., 2022) utilizes BeatNet activations and proposes the jump-back reward strategy, a semi-Markov inference method, to reduce computation. 3) **IBT** (Oliveira et al., 2010) is a signal processing based method that uses onset strength to select an agent with the most correct beat position hypothesis out of multiple agents. 4) **Böck FF** Böck et al. (2014) utilizes an RNN and a forward algorithm for beat tracking. 5) **BEAST** (Chang and Su, 2024) employs a streaming Transformer and a forward algorithm for joint beat and downbeat tracking, achieving the best performance over existing state-of-the-art models on the GTZAN benchmark. Among the reported methods, IBT and Böck FF only perform beat tracking and do not provide downbeat results.

It is also important to mention that certain prior studies, such as Beast (Chang and Su, 2024), present their results by incorporating multiple hop-size look-ahead steps in addition to their real-time online performance. While these look-ahead steps enhance the performance of rhythm analysis systems, they introduce significant delays and make the models non-causal. To ensure a fair and consistent comparison among online models, we only compare the fully online performance of all models.

To better put online music rhythm analysis methods in context, we also compare with two state-of-the-art offline rhythm analysis models. They include 1) **Transformers** (Zhao et al., 2022) model that uses a transformer encoder for estimating the activations and dynamic Bayesian Networks (DBN) for decisions, and 2) **SpecTNT-TCN** (Hung et al., 2022) that leverages a combination of Temporal Convolutional Networks (TCN) and SpecTNT (Lu et al., 2021), which integrates spectral and temporal information, to calculate activations and a DBN block for decisions.

For the two challenging scenarios, isolated singing voices and non-percussive music, we evaluate the two proposed adaptation methods. **AF** represents the first adaptation approach illustrated in Figure 2, where the auxiliary branch (right) is initialized with the frozen weights of the BeatNet+ generic model, and the main branch (left) undergoes training on the particular music arrangement and is used for inference. **GF** represents the second adaptation approach illustrated in Figure 3, involving fine-tuning a pre-trained model for specific tasks by adaptation of the input data over time.

To assess the effectiveness of the adaptation approaches, we also present results for the same models trained from scratch for the specific tasks, without leveraging the adaptation techniques. These models are referred to as **AF-scratch** and **GF-scratch**, respectively. In particular, **AF-scratch** uses the auxiliary branch structure and training data, but trained from scratch without initializing the auxiliary branch weights with the frozen weights of the pre-trained **BeatNet+** main branch. **GF-scratch** utilizes **GF** single branch structure, trained from the scratch and without guided fine-tuning.

For singing voice rhythm analysis, we compare with **SingNet** (Heydari et al., 2023), the current state of the art for this task. For non-percussive music rhythm analysis, no prior models are available. Thus, we compare with the state-of-the-art real-time rhythm analysis method, **BeatNet** (Heydari et al., 2021), when is trained exclusively on non-percussive music pieces.

4.4 Training Details

This section covers the training details of the BeatNet+ models for generic music rhythm analysis as well as the “auxiliary-freezing” and “guided fine-tuning” adaptation techniques for challenging scenarios.

All proposed models are trained using the Adam optimizer with a constant learning rate of 5×10^{-4} and a batch size of 40. All models employ a cross-entropy loss with logits, whose weights are set to 200 for downbeats, 60 for beats, and 1 for non-beats, accounting for their average occurrence rates across total training audio frames. The feature matching MSE loss weight for models with auxiliary training is set to $\lambda = 200$. Training batches comprise randomly selected 15-second excerpts from the training audio files.

For the BeatNet+ and BeatNet+ (Solo), AF-scratch and GF-scratch, all weights and biases are randomly initialized. In contrast, the AF model only initializes its main branch randomly, while its auxiliary branch is initialized as the pre-trained main branch of BeatNet+. Similarly, the GF model is also initialized as the pre-trained main branch of BeatNet+.

Note that for all external comparison methods, their pre-trained models are utilized. However, for non-percussive music rhythm analysis, the benchmark BeatNet model is trained on non-percussive audio with the training specifics of the original BeatNet model.

4.5 Results and Discussions

In this section, we present our evaluation results for various scenarios on the GTZAN dataset. We report the performance of the proposed model and adaptation techniques for generic music, isolated singing voices, and non-percussive music rhythm analysis.

4.5.1 Results on Generic Music

Table 2 compares the performance of online rhythm analysis methods as well as two offline methods for generic music. We can see that the proposed BeatNet+ outperforms all the other online methods on both beat tracking and downbeat tracking F1 scores, while maintaining competitive latency and RTF. Regarding computational complexity, the Novel 1D model achieves the lowest RTF, thanks to its utilization of an exceptionally lightweight inference approach. The F1 score improvement from BeatNet+ (Solo) to BeatNet+, es-

Method	Metrics (Performance on Full Mixtures)			
	Beat F1↑ (70ms)	Downbeat F1↑ (70ms)	Latency↓ (ms)	RTF↓
		Online Models		
BeatNet+	**80.62**	**56.51**	**20**	0.08
BeatNet+ (Solo)	78.43	49.74	**20**	0.08
BeatNet (Heydari et al., 2021)	75.44	46.69	**20**	0.06
Novel 1D (Heydari et al., 2022)	76.47	42.57	**20**	**0.02**
IBT (Oliveira et al., 2010)	68.99	—-	23	0.16
Böck FF (Böck et al., 2014)	74.18	—-	46	0.05
Beast (Chang and Su, 2024)	80.04	52.23	46	0.40
		Offline Models		
Transformers(Zhao et al., 2022)	88.5	71.4	—-	—-
SpecTNT-TCN (Hung et al., 2022)	88.7	75.6	—-	—-

Table 2: Results of online rhythm analysis evaluation for generic music and offline state-of-the-art references, showcasing F1 scores in % with a tolerance window of 70 ms, latency, and RTF for the GTZAN dataset.

pecially on downbeat tracking, highlights the benefit of using the auxiliary branch during the training process and leveraging the latent-matching technique between the two branches; The latency and RTF do not change as BeatNet+ utilizes only one branch during inference. Finally, BeatNet+ (Solo) improves over BeatNet on both beat and downbeat F1 scores.

In the comparative analysis between BeatNet+ and Beast, BeatNet+ demonstrates a marginal advantage in beat tracking and a significant superiority in downbeat tracking. Noteworthy is the fact that the latency and RTF of BeatNet+ models are more than two times and nearly seven times shorter than those of the Beast model, making them more convenient for real-time and low-resource applications. The main reason for its substantially reduced computational cost lies in its utilization of a source-efficient light 1D CRNN model, in contrast to the inclusion of streaming transformers used in Beast.

To assess system performance across various genres, we present the beat and downbeat F1 scores achieved by the top-performing method, BeatNet+, across all GTZAN genres in Figure 4. A comparative analysis of the reported box plots reveals notable variations in model performance for different genres. Specifically, the model's best overall performance is observed for Disco and Hip-hop; This is potentially attributed to the presence of strong percussive and harmonic cues and their more straightforward rhythmic patterns. Conversely, genres like Classical and Jazz demonstrate below-average model performance, potentially due to the diverse musical characteristics and intricate rhythmic patterns inherent to these genres.

Interestingly, some genres show contrasting performance between beat tracking and downbeat tracking. Specifically, Reggae receives one of the best beat tracking performance but the second-worst downbeat tracking performance with the widest range across different pieces. This suggests that, while the percussive and harmonic elements of Reggae are ample for beat tracking, they are not sufficient for distinguishing between beats and downbeats. This phenomenon is attributed to the presence of a substantial amount of syncopation and frequently used off-beat rhythmic patterns such as “One-drop”, “Steppers” and “Rockers” in Reggae. Similarly, Jazz and Blues also show large performance disparity between beat and downbeat tracking, attributable to the prevalent use of styles such as the “Swing feel” within these genres[2].

4.5.2 Results on Singing Voices

Rhythm analysis of isolated singing voices is the most challenging task among all discussed in this work. The first row of Figure 5 compares the F1 scores of the proposed model with different adaptation strategies against SingNet Heydari et al. (2023), the state-of-the-art singing voice rhythm analysis model, on singing stems of the GTZAN dataset. According to the figure, GF delivers the best performance for beat tracking by a significant improvement of 14.58% and 13.27% over the SingNet model for $T = 70ms$ and $T = 200ms$ tolerances, respectively. For downbeat tracking, AF outper-

[2] *Syncopation:* Irregular drum patterns by accenting weak beats commonly not emphasized, and by omitting or displacing notes, such as downbeats and upbeats, in a 4/4 meter. *One drop:* is a prominent drum set rhythm in reggae, differing from the typical backbeat by emphasizing the kick on beats 2 and 4 instead of 1 and 3. *Steppers:* follows the “four on the floor” pattern, featuring the kick drum hitting on all four downbeats in each measure. *Rockers:* a reggae beat in which the kick drum is on 1 and 3, while the snare is on beats 2 and 4 in a 4/4 meter. *Swing feel:* a specific type of syncopation that emphasizes the off-beat, giving the music a bouncy, lively feel (Morena, 2021).

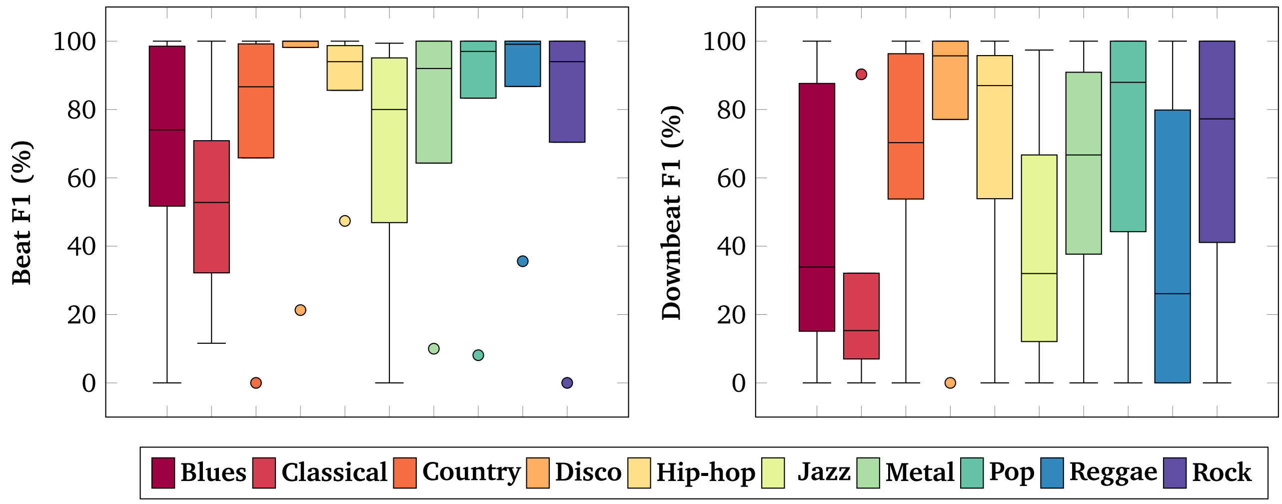

Figure 4: F1 scores for beat tracking and downbeat tracking of the BeatNet+ model across diverse genres within the GTZAN dataset.

forms SingNet by 2.43% and 0.51% for $T = 70ms$ ms and $T = 200ms$, tolerances. A more significant improvement in beat tracking accuracy compared to downbeat tracking suggests that the proposed models enhance acoustic modeling more effectively than capturing higher-level semantic modeling.

Comparing the adaptation models with the same BeatNet+ structures trained from scratch, GF outperforms GF-scratch significantly for beat tracking across both tolerances. However, it marginally underperforms GF-scratch for downbeat tracking. On the other hand, AF outperforms AF-scratch, for downbeat tracking while underperforming AF-scratch for beat detection. The aforementioned records indicate that for singing voice rhythm analysis, guided fine-tuning and auxiliary freezing techniques are effective for beat and downbeat tracking, respectively. However, there is no optimal joint model for both tasks.

4.5.3 Results on Non-Percussive Music

Rhythm analysis of non-percussive music is another challenging task. The plots on the second row of Figure 5 compare the performance of the proposed BeatNet+ model with different adaptation strategies against the BeatNet model on GTZAN pieces after removing the drums. As mentioned earlier, for this comparison, the BeatNet model is trained on the same data as the proposed models, i.e., non-percussive parts of the training set from scratch. According to the results, AF delivers the best performance for both beat and downbeat tracking among all models with a significant improvement of 8.88% and 8,19% for $T = 70$ and 10.55% and 12.85% for $T = 200$ over the baseline BeatNet model.

Comparing AF with AF-scratch underscores the impact of the auxiliary freezing technique on non-percussive music rhythm analysis. Disabling auxiliary freezing results in a notable downgrade in model performance, shifting it from being the best across all models to the overall worst. However, comparing GF with GF-scratch reveals that guided fine-tuning offers similar performance for non-percussive rhythm analysis.

Also, We acknowledge that the rhythm analysis performance for non-percussive and isolated singing voices may be impacted by residual signals and data leakage, resulting from utilizing source separation techniques to extract music stems for training and evaluation. However, prior studies such as Heydari et al. (2021) have shown that this effect is negligible, as evidenced by comparing their model performances on music pieces with pure stems versus separated ones. Importantly, a fair comparison is ensured by using the same datasets for all reported models.

5. Conclusion

This paper presents BeatNet+, a cutting-edge online rhythm analysis model that significantly advances the state of the art in real-time music rhythm analysis. By incorporating an auxiliary branch regularization mechanism and employing innovative adaptation strategies, BeatNet+ demonstrates outstanding performance across various music scenarios, including generic music pieces, isolated singing voices, and non-percussive audio tracks. Additionally, we release the rhythmic annotations of MUSDB and URSing datasets, enabling them to be utilized for music rhythm analysis as well as revised annotations of RWC Jazz, Pop and Royalty-free along with this work.

Acknowledgements

This work has been supported by the XXXXXX grant XXXXX.

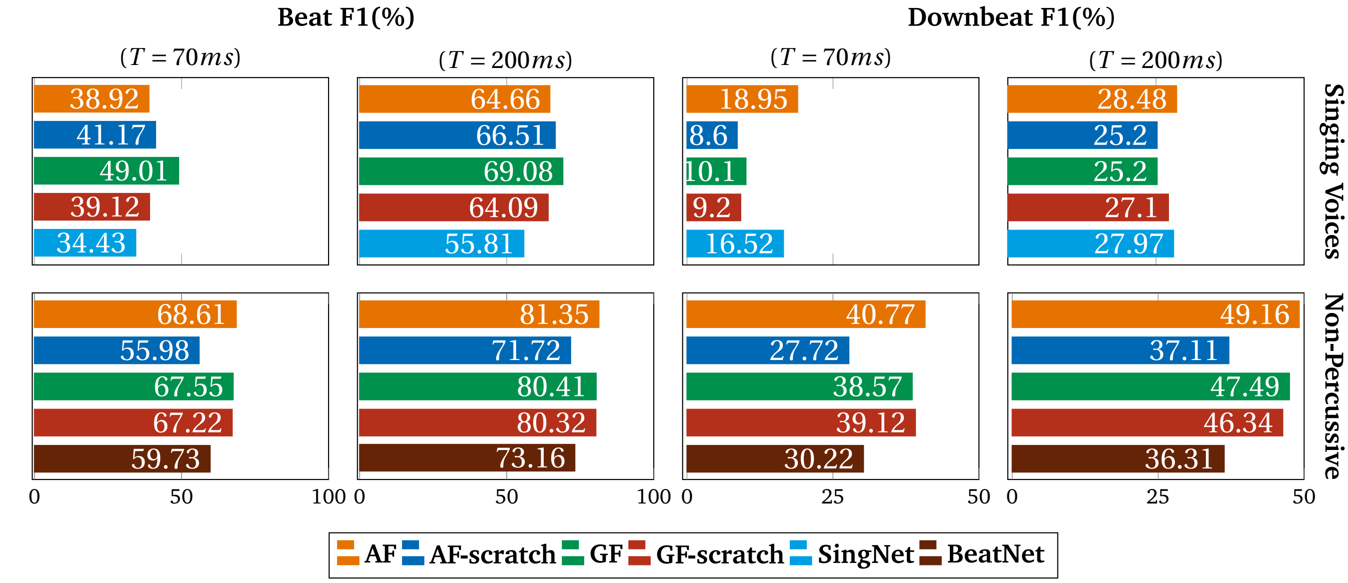

Figure 5: F1 scores of online rhythm analysis models on singing voices (top row) and non-percussion music (bottom row) with two tolerance windows, 70 ms and 200 ms.

References

Bain, M. N. (2008). Real time music visualization: A study in the visual extension of music. Master's thesis, The Ohio State University.

Bégel, V., Seilles, A., and Dalla Bella, S. (2018). Rhythm workers: A music-based serious game for training rhythm skills. *Music & Science*, 1:2059204318794369.

Bi, T., Fankhauser, P., Bellicoso, D., and Hutter, M. (2018). Real-time dance generation to music for a legged robot. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1038–1044. IEEE.

Böck, S. and Davies, M. E. (2020). Deconstruct, analyse, reconstruct: How to improve tempo, beat, and downbeat estimation. In *In Proc. International Society for Music Information Retrieval Conference (ISMIR)*, pages 574–582.

Böck, S., Krebs, F., and Widmer, G. (2014). A multi-model approach to beat tracking considering heterogeneous music styles. In *In Proc. International Society for Music Information Retrieval Conference (ISMIR)*, pages 603–608. Citeseer.

Böck, S., Krebs, F., and Widmer, G. (2016). Joint beat and downbeat tracking with recurrent neural networks. In *ISMIR*, pages 255–261. New York City.

Böck, S. and Schedl, M. (2011). Enhanced beat tracking with context-aware neural networks. In *Proc. Int. Conf. Digital Audio Effects*, pages 135–139.

Chang, C.-C. and Su, L. (2024). BEAST: Online joint beat and downbeat tracking based on streaming transformer. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

Chang, H.-J., Yang, S.-W., and Lee, H.-Y. (2022). Distilhubert: Speech representation learning by layer-wise distillation of hidden-unit BERT. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE.

Chen, S., Wang, C., Chen, Z., Wu, Y., Liu, S., Chen, Z., Li, J., Kanda, N., Yoshioka, T., Xiao, X., et al. (2022). WavLM: Large-scale self-supervised pre-training for full stack speech processing. *IEEE Journal of Selected Topics in Signal Processing*, 16(6):1505–1518.

Chiu, C.-Y., Müller, M., Davies, M. E. P., Su, A. W.-Y., and Yang, Y.-H. (2023). Local periodicity-based beat tracking for expressive classical piano music. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 31:2824–2835.

Chiu, C.-Y., Su, A. W., and Yang, Y.-H. (2021). Drum-aware ensemble architecture for improved joint musical beat and downbeat tracking. *IEEE Signal Processing Letters*, 28:1100–1104.

Cliff, D. (2000). Hang the dj: Automatic sequencing and seamless mixing of dance-music tracks. *Hp Laboratories Technical Report Hpl*, 104.

Davies, M. E. and Böck, S. (2019). Temporal convolutional networks for musical audio beat tracking. In *2019 27th European Signal Processing Conference (EUSIPCO)*, pages 1–5. IEEE.

Davies, M. E., Brossier, P. M., and Plumbley, M. D. (2005). Beat tracking towards automatic musical accompaniment. In *Audio Engineering Society Convention 118*.

Davis, A. and Agrawala, M. (2018). Visual rhythm and beat. *ACM Transactions on Graphics (TOG)*, 37(4):1–11.

De Clercq, T. and Temperley, D. (2011). A corpus analysis of rock harmony. *Popular Music*, 30(1):47–70.

Défossez, A. (2021). Hybrid spectrogram and waveform source separation. In *In Proc. International Society for Music Information Retrieval Conference (ISMIR), 2021 Workshop on Music Source Separation*.

Desblancs, D., Lostanlen, V., and Hennequin, R. (2023). Zero-Note Samba: Self-supervised beat tracking. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*.

Ellis, D. P. (2007). Beat tracking by dynamic programming. *Journal of New Music Research*, 36(1):51–60.

Elowsson, A. (2016). Beat tracking with a cepstroid invariant neural network.

Eskimez, S. E., Maddox, R. K., Xu, C., and Duan, Z. (2019). Noise-resilient training method for face landmark generation from speech. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:27–38.

Eyben, F., Weninger, F., Ferroni, G., and Schuller, B. (2013). Tempo estimation and beat tracking with long short-term memory neural networks and comb-filters. *Universität Augsburg, 2013*.

Federgruen, A. and Tzur, M. (1991). A simple forward algorithm to solve general dynamic lot sizing models with n periods in 0 (n \log n) or 0 (n) time. *Management Science*, pages 909–925.

Gkiokas, A. and Katsouros, V. (2017). Convolutional neural networks for real-time beat tracking: A dancing robot application. In *In Proc. International Society for Music Information Retrieval Conference (ISMIR)*, pages 286–293.

Gkiokas, A., Katsouros, V., and Carayannis, G. (2012). Reducing tempo octave errors by periodicity vector coding and svm learning. In *ISMIR*, pages 301–306.

Goto, M. (2001). An audio-based real-time beat tracking system for music with or without drum-sounds. *Journal of New Music Research*, 30(2):159–171.

Goto, M. (2004). Development of the rwc music database. In *Proc. of the 18th International Congress on Acoustics, 2004*, pages 553–556.

Goto, M., Hashiguchi, H., Nishimura, T., and Oka, R. (2002). Rwc music database: Popular, classical and jazz music databases. In *In Proc. International Society for Music Information Retrieval Conference (ISMIR)*.

Goto, M. and Muraoka, Y. (1999). Real-time beat tracking for drumless audio signals: Chord change detection for musical decisions. *Speech Communication*, 27(3-4):311–335.

Gouyon, F., Klapuri, A., Dixon, S., Alonso, M., Tzanetakis, G., Uhle, C., and Cano, P. (2006). An experimental comparison of audio tempo induction algorithms. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(5):1832–1844.

Greenlees, M. (2020). Beat tracking with autoencoders. *https://doi.org/10.5281/zenodo.4091524, doi:https://doi.org/10.5281/zenodo.4091524*.

Hainsworth, S. W. and Macleod, M. D. (2004). Particle filtering applied to musical tempo tracking. *EURASIP Journal on Advances in Signal Processing*, 2004:1–11.

Heydari, M., Cwitkowitz, F., and Duan, Z. (2021). Beatnet: CRNN and particle filtering for online joint beat downbeat and meter tracking. *In Proc. International Society for Music Information Retrieval Conference (ISMIR)*.

Heydari, M. and Duan, Z. (2021). Don't look back: An online beat tracking method using RNN and enhanced particle filtering. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE.

Heydari, M. and Duan, Z. (2022). Singing beat tracking with self-supervised front-end and linear transformers. *In Proc. International Society for Music Information Retrieval Conference (ISMIR)*.

Heydari, M., McCallum, M., Ehmann, A., and Duan, Z. (2022). A novel 1D state space for efficient music rhythmic analysis. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE.

Heydari, M., Wang, J.-C., and Duan, Z. (2023). SingNet: A real-time singing voice beat and downbeat tracking system. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE.

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

Hung, Y.-N., Wang, J.-C., Song, X., Lu, W.-T., and Won, M. (2022). Modeling beats and downbeats with a time-frequency transformer. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE.

Katharopoulos, A., Vyas, A., Pappas, N., and Fleuret, F. (2020). Transformers are RNNs: Fast autoregressive transformers with linear attention. In *International Conference on Machine Learning*, pages 5156–5165. PMLR.

Kim, Y. and Rush, A. M. (2016). Sequence-level knowledge distillation.

Krebs, F., Böck, S., and Widmer, G. (2013). Rhythmic pattern modeling for beat and downbeat tracking in musical audio. In *Ismir*, pages 227–232. Citeseer.

Krebs, F., Böck, S., and Widmer, G. (2015). An efficient state-space model for joint tempo and meter tracking. In *ISMIR*, pages 72–78.

Li, B., Wang, Y., and Duan, Z. (2021). Audiovisual singing voice separation.

Lu, W.-T., Wang, J.-C., Won, M., Choi, K., and Song, X. (2021). SpecTNT: A time-frequency transformer for music audio. *In Proc. International Society for Music Information Retrieval Conference (ISMIR)*.

Marchand, U. and Peeters, G. (2015). Swing ratio estimation. In *Digital Audio Effects 2015 (Dafx15)*.

Masri, P. (1996). *Computer Modeling of Sound for Transformation and Synthesis of Musical Signals*. PhD thesis, University of Bristol.

Meier, P., Krump, G., and Müller, M. (2021). A real-time beat tracking system based on predominant local pulse information. *Demos and Late Breaking News of the International Society for Music Information Retrieval Conference (ISMIR)*.

Morais, G., EP Davies, M., Queiroz, M., and Fuentes, M. (2023). Tempo vs. pitch: Understanding self-supervised tempo estimation. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE.

Morena, E. (2021). *A Creative Exploration of Techniques Employed in Pop/Rock Drum Patterns (1965–1992): A dissertation with supporting audio and video recordings*. PhD thesis.

Mottaghi, A., Behdin, K., Esmaeili, A., Heydari, M., and Marvasti, F. (2017). OBTAIN: Real-time beat tracking in audio signals. *International Journal of Signal Processing Systems*.

Oliveira, J. L., Gouyon, F., Martins, L. G., , and Reis, L. P. (2010). IBT: A real-time tempo and beat tracking system. In *Proc. of the 11th Intl. Conf. on Music Information Retrieval (ISMIR)*, pages 291–296.

Rafii, Z., Liutkus, A., Stöter, F.-R., Mimilakis, S. I., and Bittner, R. (2017). The MUSDB18 corpus for music separation.

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

Schloss, W. A. (1985). *On the Automatic Transcription of Percussive Music–From Acoustic Signal to High-level Analysis*. Stanford University.

Shiu, Y. and Kuo, C.-C. J. (2007). A modified kalman filtering approach to on-line musical beat tracking. In *IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP*. IEEE.

Steinmetz, C. J. and Reiss, J. D. (2021). WaveBeat: End-to-end beat and downbeat tracking in the time domain.

Tsunoo, E., Kashiwagi, Y., Kumakura, T., and Watanabe, S. (2019). Transformer ASR with contextual block processing. In *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 427–433. IEEE.

Tzanetakis, G. and Cook, P. (2002). Musical genre classification of audio signals. *IEEE Transactions on speech and audio processing*, 10(5):293–302.

Wu, Y.-K., Chiu, C.-Y., and Yang, Y.-H. (2022). Jukedrummer: Conditional beat-aware audio-domain drum accompaniment generation via transformer VQ-VAE.

Zhao, J., Xia, G., and Wang, Y. (2022). Beat transformer: Demixed beat and downbeat tracking with dilated self-attention. *In Proc. International Society for Music Information Retrieval Conference (ISMIR)*.

Zheng-qing, C. and Jian-hua, H. (2005). A comparative study between time-domain method and frequency-domain method for identification of bridge flutter derivatives. *Engineering Mechanics*, 22(6):127–133.