LEARNING HIERARCHICAL METRICAL STRUCTURE BEYOND MEASURES

Junyan Jiang[1,2]   **Daniel Chin**[1,2]   **Yixiao Zhang**[1,3]   **Gus Xia**[1,2]

[1] Music X Lab, NYU Shanghai   [2] MBZUAI   [3] Centre for Digital Music, QMUL

`jj2731@nyu.edu, daniel.chin@nyu.edu, yixiao.zhang@qmul.ac.uk, gxia@nyu.edu`

ABSTRACT

Music contains hierarchical structures beyond beats and measures. While hierarchical structure annotations are helpful for music information retrieval and computer musicology, such annotations are scarce in current digital music databases. In this paper, we explore a data-driven approach to automatically extract hierarchical metrical structures from scores. We propose a new model with a Temporal Convolutional Network-Conditional Random Field (TCN-CRF) architecture. Given a symbolic music score, our model takes in an arbitrary number of voices in a beat-quantized form, and predicts a 4-level hierarchical metrical structure from downbeat-level to section-level. We also annotate a dataset using RWC-POP MIDI files to facilitate training and evaluation. We show by experiments that the proposed method performs better than the rule-based approach under different orchestration settings. We also perform some simple musicological analysis on the model predictions. All demos, datasets and pre-trained models are publicly available on Github [1].

1. INTRODUCTION

Music contains rich structures at different levels, and the structure annotations play an important role in music understanding [1, 2] and generation [3, 4]. Progress has been made in music structure analysis on certain levels, like beat tracking [5–7], downbeat detection [8–11] and part segmentation [12–15]. These levels of structures are often inter-connected with each other, and can be described in a hierarchical way. For example, a part may contain several sections, each containing several measures. Measures can be further decomposed into beats.

Several views of hierarchical music structures are formally discussed in the Generative Theory of Tonal Music (GTTM) [16], including the grouping structure, the metrical structure, the time-span tree, and the prolongational tree, each focusing on different music properties (i.e., the grouping structure focuses more on melodic grouping

Figure 1. The hierarchical metrical structure of the beginning of RWC-POP No. 001. Only the main melody is shown. Despite the pick-up bar, these 4 measures are grouped into 2 hypermeasures of length 2, which are further grouped into 1 hypermeasure of length 4.

while the metrical structure focuses on rhythmic patterns). Among these structures, we choose the metrical structure as the topic for two reasons: (1) For polyphonic music, the metrical structures of different voices are usually compatible, building up a common song-level metrical structure. This property makes data annotation easier and provides opportunities for self-supervision; (2) Some low-level metrical structures (e.g., beats and downbeats) are already annotated in music scores and most MIDI datasets, which can be a helpful source of supervision. In this paper, we will focus our analysis on pop songs, which often contain a well-defined hierarchy of metrical structures [17].

Our main goal is to infer the high-level metrical structures given low-level ones like beats and downbeats. The hierarchy of the metrical structure is created by recursively grouping lower metrical units into upper ones (see Figure 1 for an example). We call the grouping of measures (or other larger metrical units) a *hypermeasure*, and the number of units that form the group as its *hypermeter* [18, 19]. While some properties of beats and measures can be generalized to upper-level metrical structures, there are still many differences to take into account. Similar to the meter, the hypermeter of each layer tends to stay the same to maintain a regular rhythmic pulse, but it is not uncommon to see hypermeter changes in a piece, as shown in Figure 4. Such changes occur more often than low-level meter changes since listeners are less sensitive to long-term rhythmic regularity. Another major difference is that the decision of upper-level metrical structures requires a longer context in the time domain compared to downbeat or beat tracking.

To resolve these issues, we design a new model that contains a Temporal Convolutional Network (TCN) front-end for metrical level prediction and a Conditional Random Field (CRF) decoder for joint metrical structure decoding. We design the transition of CRF hidden states to allow hypermeter changes with some penalties. To han-

---

[1] https://github.com/music-x-lab/Hierarchical-Metrical-Structure

dle polyphony, the model takes an arbitrary number of voices/tracks as input, and predicts a confidence score for each track. The final prediction is a weighted average of the results from all tracks. We annotated 70 songs from the RWC-POP dataset and used them for model training and evaluation. We conduct experiments under different orchestration setups with results shown in section 4.4.

2. RELATED WORK

2.1 Downbeat and Meter Tracking

Downbeat tracking for raw audio is a well-studied task with many promising results. Krebs et al. [8] use particle filters to find the downbeats, yielding a 2-level metrical structure. Durand et al. [9] use an ensemble of convolutional networks to locate downbeats robustly. Fuentes et al. [10] use skip-chain CRF and deep learning to track downbeats, noting that longer-term musical contexts can better inform downbeat tracking. The reader is referred to Gouyon and Dixon's review [11] for non-symbolic low-level metrical analyzers before 2005.

Notably, locating the beats and downbeats for symbolic music is not a trivial task. Kostek et al. [20] apply neural networks and rough sets to a polyphonic symbolic piece and classify whether each note is accented or not, yielding a 2-level metrical structure (beat and downbeat). Chuang and Su [21] use various RNNs to classify each timestep in a piano roll into non-beat, beat, or downbeat.

Another related task is time signature detection. Benoit [22] uses symbolic-level auto-correlation to obtain a 4-level metrical structure, and ultimately extracts the meter. More auto-correlative methods [23, 24] share a similar logic, since note onsets usually display periodicity in every measure. More recently, inner metric analysis was also used to infer the time signature by Haas et al. [25].

2.2 Music Segmentation

The task of music segmentation is to infer musically meaningful section or part boundaries from the music content. Audio-based music segmentation is usually achieved by detecting similarity or repetition of the audio spectral features. McFee and Ellis [13] evaluate inter-time frame similarity and use spectral clustering to obtain a multi-level segmentation of music. Salamon et al. [14] and McCallum [15] replace traditional features with pre-trained deep embeddings to estimate timbre and harmonic similarity. Tralie and McFee [26] fuse multiple similarity metrics for better prediction results. Ullrich et al. [27] train fully supervised CNN on a segment-annotated dataset to detect segment boundaries. See Dannenberg and Goto's review [28] and the 2010 SOTA report by Paulus et al. [29] to learn more about audio-based music segmentations.

Segmentation of symbolic music relies more on domain knowledge of music composition. Van der Werf and Hendriks [30] restate GTTM grouping rules in terms of Optimality Theory (OT) and design a Prolog program to find an optimal parse for short monophonic pieces. Dai et al. [31]

identify phrases as units of melodic repetition by minimizing the Structural Description Length (SDL) for the entire piece, and then extract a 2-layer hierarchy.

2.3 Hierarchical Structure Analysis

The Generative Theory of Tonal Music (GTTM) [16] discusses several views of the hierarchical music structures, but GTTM does not describe how to realize an analyzer computationally. Various efforts have been made to mechanize GTTM. For example, Jones et al. [32] use a rule-based expert system to obtain a 6-level metrical structure for monophonic pieces, satisfying all GTTM's well-formedness rules while following Povel's grid theory [33]. Rosenthal's Machine Rhythm [34] ventures into the polyphonic domain and uses rule-based methods to extract a 3-level rhythmic annotation for MIDI input. Temperley and Sleator [35] use a preference-rule approach and dynamic programming to obtain the optimal parse. Hamanaka et al. [36] propose the Automatic Time-span Tree Analyzer (ATTA) for structural analysis of 8-bar monophonic scores. Temperley [37] use Bayesian reasoning to jointly analyze metrical, harmonic, and stream structures. Wojcik and Kostek [18] use rule-based methods to retrieve hypermetric rhythm from only the melody.

Machine learning models are also used for hierarchical structure analysis. Hamanaka et al. propose Deep-GTTM [38, 39] which is a fully automatic analyzer that uses a neural network to predict the applicability of GTTM rules for each note, yielding a 5-level metrical structure for 8-bar monophonic scores.

There are some issues when applying previous works to large polyphonic MIDI databases. Most systems work only for monophonic music or music with limited polyphony. Also, they often work in a short context, usually up to 8 bars.

3. METHODOLOGY

3.1 Problem Setting

We first formally define the metrical structure prediction task for this paper. Assume we have a music score with T voices (or tracks, in the sense of MIDI files) $\mathbf{m}_1, ..., \mathbf{m}_T$. We also have a list of pre-annotated downbeats $d_1, ..., d_N$ for the music. The aim is to assign hierarchical metrical labels $l_i \in \{0, 1, ..., L\}$ to each downbeat d_i where L is the total number of layers we want to build beyond measures. Each label serves as the level of the metrical boundary at d_i. $l_i = l$ means d_i serves as a metrical boundary of all levels for the first l levels beyond measures. Specially, $l_i = 0$ means d_i serves only as a measure boundary but not any metrical boundary beyond measures.

If we use GTTM's metrical structure notation, we can use $(l_i + 1)$ dots to represent a metrical boundary level of l_i. For the example in figure 1, the first 4 measures (excluding the pickup measure) would have labels $l_1 = 2, l_2 = 0, l_3 = 1, l_4 = 0$.

We can also introduce the following notations.
Hypermeasures: A hypermeasure of level l is an interval
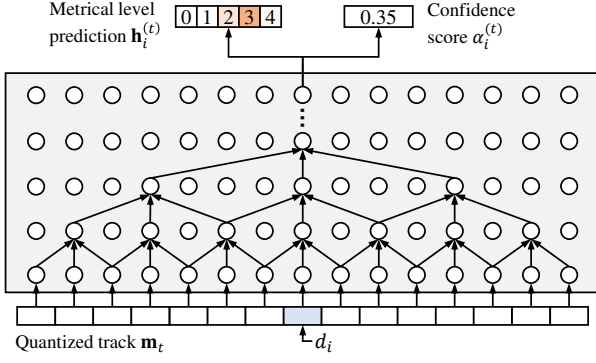
Figure 2. The architecture of the neural network.

between two downbeats d_i and d_j where $l_i \geq l, l_j \geq l$ and $l_k < l$ for all $k = i + 1...j - 1$. In other words, any $l_k \geq l$ serves as a separator of a level-l hypermeasure. Specially, level-0 hypermeasures are just measures.

Hypermeters: A hypermeter is the generalization of meters by counting how many level-$(l-1)$ hypermeasures are in a level-l hypermeasure. Since a binary structure is the most commonly used [19, 40], we assume a general hypermeter of 2 in all levels, with a few exceptions that cause *binary irregularity*.

3.2 Temporal Convolutional Network

Temporal Convolutional Networks (TCNs) have been proven an effective model for beat, downbeat, and tempo tracking [7, 41, 42]. We believe it is useful for general metrical structure analysis for its unique property we will mention below. We mainly reference [43] for the design of the TCN, but we made it non-causal similar to [7]. Each TCN block contains 8 sequential layers. The first 4 layers are a dilated convolutional layer with kernel size 3 and 256 channels, a batch normalization layer, a Rectified Linear Unit (ReLU) activation layer, and a dropout layer. The next 4 layers repeat this configuration. There is also a residual component that adds a linear transformation of the input to the block output, allowing shortcut connections.

Our model uses 6 TCN blocks sequentially. Each block multiplies the dilation by 2, starting from 1 at block 1, resulting in an exponentially growing context range for each layer. This allows the model to capture long-term context and more importantly, integrate prior knowledge about binary metrical structure into the network. The model input contains the piano roll and the onset roll of a track quantized into a 16th note level. Under a 4/4 meter, the dilations of the convolutional layers in each block are therefore 1/4 beat, 1/2 beat, 1 beat, 2 beats, 1 measure and 2 measures, respectively. This encourages the convolutional layers to capture more musically meaningful context for binary metrical structures.

For each track $\mathbf{m}_t$ in a song, we first feed them into the TCN blocks to get the features for each time step, and then discard the time steps that do not correspond to any downbeat. We use linear layers to project the features into a metrical level prediction $\mathbf{h}_i^{(t)}$ and a confidence score $\alpha_i^{(t)}$.
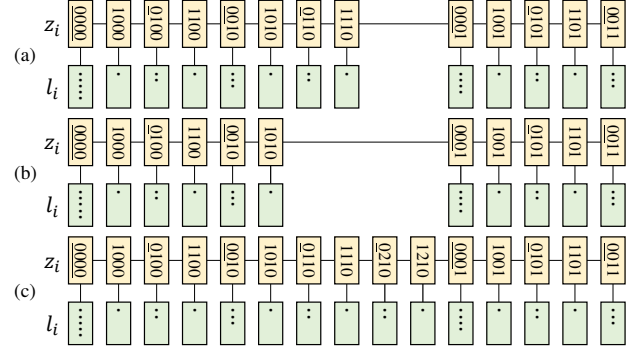
Figure 3. Examples of the CRF hidden variables z_i (shown as a L-digit number $z_i^{(1)}...z_i^{(L)}$) and the corresponding metrical boundary levels l_i when $L = 4$. (a) Binary regularity is satisfied. (b) A level-1 hypermeasure is deleted from (a). (c) A level-1 hypermeasure is inserted into (a). Both (b) and (c) are examples of binary irregularity.

Each prediction $\mathbf{h}_i^{(t)}$ is a vector of size $(L+1)$ for the labels $0...L$. The final prediction of l_i is the weighted average of all predictions $\mathbf{h}_i^{(t)}$ on the same time step weighted by their confidence scores:

$$a_i^{(t)} := \exp \alpha_i^{(t)} / \sum_{t'} \exp \alpha_i^{(t')} \tag{1}$$

$$\mathbf{p}_i := \sum_t a_i^{(t)} \text{Softmax}(\mathbf{h}_i^{(t)}) \tag{2}$$

3.3 Conditional Random Fields

Conditional Random Fields (CRFs) have been widely used in downbeat tracking [10, 44] to enforce the regularity of the decoded downbeat patterns. Inspired by this, we also use a structured prediction method for decoding. The major differences are that this model needs to predict a hierarchy of $L = 4$ layers of metrical structures jointly.

The state space of hierarchical metrical structure can be complex and ambiguous. To make it simple, we restrict our model to accept a hypermeter of 1, 2 or 3 on any level. In the sense of transformational grammar, a level-l hypermeter of 1 can be constructed by deleting some level-$(l-1)$ hypermeasures from a deep structure with binary regularity, and a hypermeter of 3 can be constructed by inserting some level-$(l-1)$ hypermeasures (see figure 3 for an example). A hypermeter of 4 or more is not allowed and needs to be decomposed into multiple metrical levels (e.g., $4 = 2 + 2$).

We design the linear CRF with a joint state space $z_i = (z_i^{(1)}, ..., z_i^{(L)})$ where each $z_i^{(l)} \in \{0, 1, 2\}$ corresponds to the current hypermeasure position at level l, i.e., the number of complete level-$(l-1)$ hypermeasures in this level-l hypermeasure up to the current time step. It can be seen as a generalization of the beat position in [10]. A state $z_i^{(1...l)} = 0 \wedge z_i^{(l+1)} \neq 0$ denotes a metrical boundary level $l_i = l$. Specially, the highest-level metrical boundary $l_i = L$ is associated and only associated with the state $(0, ..., 0)$, and the lowest-level metrical boundary $l_i = 0$ is associated with any z_i where $z_i^{(1)} \neq 0$. See figure 3 for some concrete examples.

We manually design the transition potential function to encode the belief of binary regularity on each level. We define the transition potential matrix of a single level as

$$\mathbf{A}^{(l)} = \begin{bmatrix} \exp\left(-w_{\text{del}}^{(l)}\right) & 1 & 0 \\ 1 & 0 & \exp\left(-w_{\text{ins}}^{(l)}\right) \\ 1 & 0 & 0 \end{bmatrix} \quad (3)$$

where $A_{ij}^{(l)}$ denotes the potential of transition from hypermeasure position i to position j on level l. $w_{\text{del}}^{(l)} > 0, w_{\text{ins}}^{(l)} > 0$ are hyperparameters that controls the penalty of a level-l hypermeasure deletion and insertion respectively. Intuitively, an alternating state sequence like $0, 1, 0, 1$ on a single level satisfies binary regularity and will not be penalized. Binary irregularity by inserting (e.g., $0, 1, 2, 0, 1$) or deleting (e.g., $0, 0, 1$) states are penalized.

In a hierarchical metrical transition to a level-l metrical boundary, the hypermeasure positions of level $1...(l+1)$ are updated, and the ones above level-$(l+1)$ keep the same. Therefore, the joint transition potential is defined as

$$\phi(z_{i-1}, z_i) = \prod_{l=1}^{L} \begin{cases} A_{z_{i-1}^{(l)} z_i^{(l)}} & l \le l_i + 1 \\ \mathbb{I}[z_{i-1}^{(l)} = z_i^{(l)}] & l > l_i + 1 \end{cases} \quad (4)$$

where l_i denotes the corresponding metrical boundary level of z_i, and $\mathbb{I}[b]$ is the indicator function that returns 1 if b is true and 0 if b is false.

The emission potential function is designed as $\psi(z_i, \mathbf{p}_i) = p_{il_i}$ where p_{il_i} is the l_i-th entry of $\mathbf{p}_i$. We use Viterbi decoding to decode the optimal hidden states $z_{1..d}$ given the observations $\mathbf{p}_{1...N}$:

$$\hat{\mathbf{z}} = \arg\max_{\mathbf{z}} \psi(z_1, \mathbf{p}_1) \prod_{i=2}^{N} \phi(z_{i-1}, z_i) \psi(z_i, \mathbf{p}_i) \quad (5)$$

4. EXPERIMENTS

4.1 Datasets

The main dataset we use in model training and evaluation is the RWC-POP dataset. It contains 100 songs with aligned MIDI files. The MIDI files have beat and downbeat annotations that are mostly correct[2]. We manually annotated the 4-layer song-level metrical structure for 70 songs. We use 50 songs for training, 10 for validation, and 10 for testing. We acknowledge the ambiguity in data annotation which potentially causes bias by annotator subjectivity [45], so we publicized the annotation data and methods to facilitate discussion and future research.

All samples are quantized into 16th-note units. The binary piano roll and onset roll are extracted for each track. During training, a random 512-unit (32 measures) segment is chosen from each song, resulting in a 512×256 feature matrix (128 MIDI pitches for piano roll and 128 MIDI pitches for onset roll) for each track. To prevent overfitting, we perform label-preserving data augmentation including random pitch shift augmentation of -12 to +12 semitones and a microtiming shift up to an 8th note on the training

---

[2] 2 out of 70 songs have minor beat/downbeat annotation issues.

set. The drum track does not use pitch shift augmentation and does not share the same parameter with pitched instruments for the first convolutional layer.

4.2 Model Training

For model training, we use a mini-batch of 16. We use the Adam optimizer [46] on the cross-entropy classification loss with a learning rate 0.0001 for 100 epochs. In one epoch, we go through each augmented version of each song for 5 times. For each song, we randomly select 2 tracks and try to predict the song-level metrical structure given the weighted average of their predictions. We also apply dropout with a probability 0.5 after each convolutional layer to further suppress overfitting.

4.3 Baseline Models

While there are some existing rule-based hierarchical metrical structure analyzers [32,39,47] in previous works, they mostly focus on low-level (e.g., beat & downbeat) boundary features like note transitions, durations and local rhythmic patterns. Those features are not effective enough for metrical structures above the measure level. We here build our baseline model using the methodology from [48]. For each metrical level, we calculate the similarity matrix of piano rolls on different granularity and estimate their novelty score as observations. We use a CRF decoder as mentioned above with a different set of hyper-parameters tuned for the baseline model.

To assess the necessity of introducing metrical irregularity, We also introduce another hypothetical baseline model called the *oracle* model. The oracle model is not allowed to predict any hypermeter changes (i.e., it always assumes binary regularity) but it always performs the best possible prediction (i.e., maximal F1 score) for each level. This hypothetical model serves as an upper bound for any model that does not allow binary irregularity.

4.4 Results

Table 1 shows the performance of each model on the test split of RWC-POP songs. To perform a more systematic evaluation, we also created two difficult versions of each test song: (1) **no drums**: the drum track(s) are removed from the original song and each model is required to predict the same metrical structure without referring to any drum clues; (2) **mel. only**: all tracks except the melody track are removed. Each model is required to predict the structure by purely referring to the main melody track.

From the results, we can see that our proposed model performs better than the rule-based counterpart on all metrical levels. Since most test songs have more than 10 MIDI tracks, they provide sufficient metrical hints to both the proposed model and the rule-based model even if the drum track is removed. When we only have the melody track, both the proposed model and the rule-based model's performances are not satisfactory even on the first level beyond measure. Still, the data-driven approach shows improved performance compared to the rule-based system.
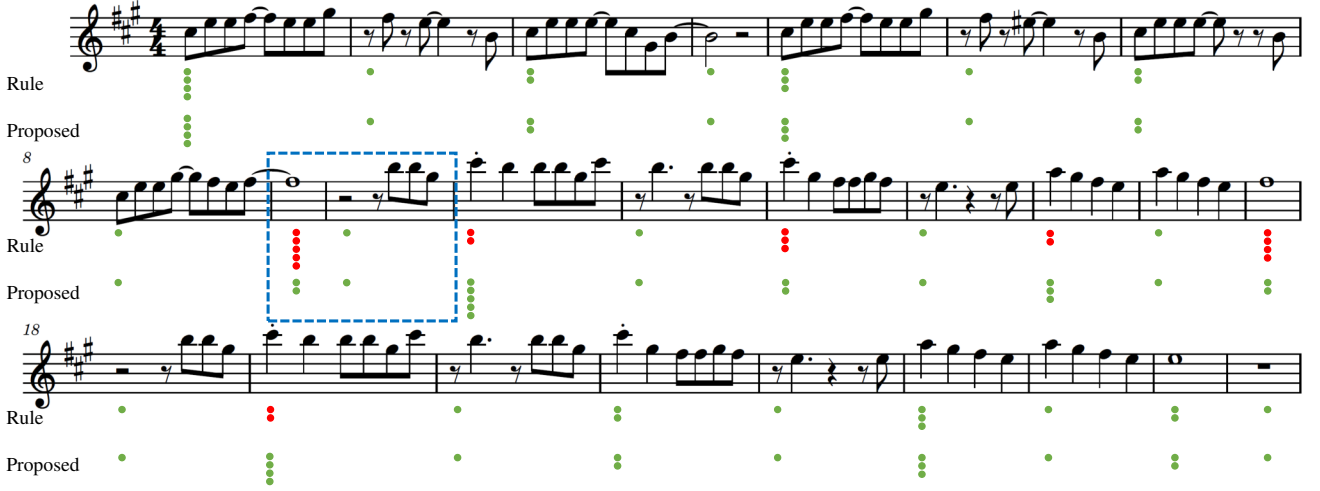
Figure 4. A case study with song RWC-POP No. 008 from the test split. The song is multi-track but we only show the main melody here. The metrical structure of the song does not satisfy binary regularity because of the 2-bar extensions in the pre-chorus (marked by a dashed blue box), causing hypermeter changes. The prediction of the proposed method aligns well with the reference. The errors in the rule-based prediction are marked in red (better viewed in color).

Model	Level 1	Level 2	Level 3	Level 4
Proposed	0.9848 ±0.0215	**0.9559** ±**0.0386**	**0.8880** ±**0.0889**	**0.6849** ±**0.1900**
Proposed w/o CRF	0.9338 ±0.0390	0.8528 ±0.0937	0.7971 ±0.1276	0.6646 ±0.0844
Rule	0.9228 ±0.0698	0.8425 ±0.1195	0.7485 ±0.1536	0.5185 ±0.2656
Oracle	0.9427 ±0.1120	0.7782 ±0.2076	0.5188 ±0.1751	0.4225 ±0.1234
Proposed (no drums)	**0.9868** ±**0.0174**	0.9519 ±0.0346	0.8803 ±0.1023	0.6611 ±0.2170
Rule (no drums)	0.9312 ±0.0660	0.8107 ±0.1568	0.7055 ±0.2008	0.4823 ±0.2239
Proposed (mel. only)	0.7413 ±0.2139	0.6253 ±0.2448	0.5551 ±0.2536	0.3808 ±0.2399
Rule (mel. only)	0.6606 ±0.1451	0.4395 ±0.1522	0.3142 ±0.1211	0.1863 ±0.1310

Table 1. Evaluated F1 scores on the test split of the RWC-POP dataset.

We observe that the proposed model is better at capturing irregular metrical structures than the rule-based approach. Figure 4 shows a cherry-picked example where binary irregularity can be found. Both the proposed model and the rule-based baseline can detect such irregularity but only the proposed model correctly tells the exact position of the hypermetrical change.

There is also a tendency for the performance to drop rapidly from lower to higher levels. We believe there are 2 main reasons. First, the higher levels have fewer positive samples, making it hard for the model to learn its semantic characteristics. Second, metrical structures on higher levels are often more ambiguous than lower ones even for human listeners. Sometimes, the highest level (level 4) needs to decide how to group parts together (e.g., verse

+ pre-chorus or pre-chorus + chorus). Different decisions are sometimes all acceptable.

4.4.1 Out of Distribution Evaluation

We also want to know whether the proposed model can be applied to MIDI files with very different orchestration setups. Such experiments are hard to perform because of the lack of ground truth annotations. We here perform a small-scale experiment on the POP909 [49] dataset. We select the first 5 songs in the dataset ordered by index and manually annotate the metrical structure [3]. POP909 is a dataset of Chinese pop songs rearranged for piano performance. Each song only has 3 tracks, i.e., a vocal track and two piano tracks, making it harder compared to RWC-POP. The results are shown in table 2.

From the results, we can see the performance degrades even when all 3 tracks are present. By case inspection, we find that the proposed model has generally satisfactory performance on 3 out of 5 songs on lower layers. However, there is one complex song [4] with multiple metrical and hypermeter changes that make all the approaches fail. Also, due to the lack of rhythmic clues (e.g., drums), a deeper understanding of the syntax and semantics of melody and chords might be required to perform musically meaningful segmentation, which we assume our model can hardly acquire on a small training set of 50 pop songs.

4.5 Confidence Score Analysis

To perform a statistical analysis of the model behavior and provide a musicological view of the model prediction, we perform model inferences on a large selection of the Lakh MIDI dataset [50]. To ensure enough accuracy of model prediction, we only select a part of the Lakh MIDI dataset

---

[3] We are aware that POP909's downbeat annotations are sometimes inaccurate and we manually fixed them.

[4] POP909 No. 005: *I Believe* by Van Fan.

Model	Level 1	Level 2	Level 3	Level 4
Proposed	**0.9084** ±**0.0896**	**0.8742** ±**0.1101**	**0.6470** ±**0.2174**	0.4930 ±0.3132
Rule	0.6818 ±0.1855	0.6625 ±0.1550	0.5163 ±0.1195	0.3446 ±0.1856
Oracle	0.8527 ±0.1735	0.7348 ±0.2763	0.5883 ±0.2493	**0.5767** ±**0.2474**
Proposed (mel. only)	0.6742 ±0.2962	0.6542 ±0.2737	0.5685 ±0.2403	0.4797 ±0.2053
Rule (mel. only)	0.6062 ±0.1034	0.3933 ±0.0275	0.2642 ±0.0546	0.1551 ±0.0305

Table 2. Evaluated F1 scores on the first 5 songs in the POP909 dataset. Mel. only denotes that the melody track is used. Otherwise, all 3 tracks (melody, bridge and piano) are used.

that has a similar orchestration compared to RWC-POP. We filter the MIDI files according to the following criteria: (1) it contains at least 6 MIDI tracks, including 1 drum track and 1 track whose name contains strings "melody" or "vocal"; (2) if multiple MIDI files' identified audio sources are the same, at most one MIDI file is kept. A filtered dataset of 3,739 MIDI files is collected.

We here evaluate the relevance of instruments and the model's predicted confidence score. Notice that the instrument program number is not a part of the model input, so the only difference comes from the rhythmic properties of their scores. We collect the unnormalized confidence scores $\alpha^{(t)}$ for each track $\mathbf{m}_t$ of different instruments, and calculate their means and standard derivations. Specially, we regard all melody tracks (identified by their names) as a new instrument and ignore its original MIDI program number. Also, we remove all tracks with too many measure-level rests (more than 1/3 of the whole song) since they trivially result in low confidence scores.

Table 3 shows the results of confidence score analysis. We can see that drums are the strongest clue for metrical structures. The melody track and many melodic instruments (e.g., guitars) also serve as useful clues. On the other hand, instruments that produce slow accompaniments (e.g., string ensemble and pads) are less preferred.

4.5.1 Drum Track Analysis

As another experiment of musicological analysis, we perform an experiment on the relation between drum notes and the metrical structure level. For simplicity, we only collect samples that a certain drum event happens exactly on the downbeat, and we collect the corresponding metrical boundary level of that downbeat. The results are shown in Table 4. While the occurrence of many drum events does not significantly change the distribution of the metrical boundary level, the crash cymbal and splash cymbal are certainly useful clues to a high-level metrical boundary. This aligns with people's perception of them since these cymbals are usually associated with a strong burst of energy, serving as an important rhythmic hint.

Track/Instrument	Confidence
Melody	1.78 ± 1.22
Drum	3.61 ± 2.58
Acoustic Grand Piano	0.35 ± 1.69
Electric Guitar (jazz)	0.75 ± 1.55
Acoustic Bass	0.33 ± 1.66
String Ensemble	0.11± 1.70
Pad (warm)	-0.86 ± 1.78

Table 3. A selected view of the means and standard derivations of the confidence score for different tracks/instruments. The melody track is identified by its name instead of the MIDI instrument. The drum track is identified by its MIDI channel number (No. 10).

Drums (%)	L-0	L-1	L-2	L-3	L-4
Any	50.00	24.71	12.06	6.21	7.02
Bass Drum	48.66	25.10	12.46	6.47	7.30
Acoustic Snare	52.27	23.28	11.19	6.27	7.00
Closed Hi Hat	51.32	25.14	11.91	5.54	6.11
Open Hi Hat	51.90	25.07	11.33	5.38	6.32
Crash Cymbal	20.97	19.13	18.58	18.94	22.38
Ride Cymbal	51.41	25.03	11.57	5.61	6.39
Splash Cymbal	34.80	22.30	16.00	12.90	14.01

Table 4. A selected view of the frequency of different drum instruments (on a downbeat) associated with different levels of metrical boundaries. L-n means level-n metrical boundary.

5. CONCLUSION AND FUTURE WORK

In this paper, we propose a data-driven approach for hierarchical metrical structure analysis of symbolic music. Our model adopts a TCN-CRF architecture and accepts an arbitrary number of voices as input. Experiments on MIDI datasets show that our model performs better than rule-based methods under different orchestration settings.

The model performance is still not satisfactory, especially for high-level metrical structures and music with very different orchestration. We assume the performance would be better if more data were annotated, but there are also other possible directions for data-driven methods. First, self-supervised or semi-supervised methods might be a helpful complement to the lack of labeled datasets. For example, a consistency loss can be used to evaluate the prediction between different voices in the same music piece. Different data augmentation strategies might also be helpful. Second, it might be useful to utilize datasets of related tasks (e.g., section labels) as a source of weak supervision. Related tasks can also be used for multi-task learning.

Other potential future works include improving the automatic analysis system of other hierarchical structures, e.g., the grouping structures. The application of hierarchical structure analysis in the audio domain is also worth exploring.

6. REFERENCES

[1] N. C. Maddage, C. Xu, M. S. Kankanhalli, and X. Shao, "Content-based music structure analysis with applications to music semantics understanding," in *Proceedings of the 12th annual ACM international conference on Multimedia*, 2004, pp. 112–119.

[2] D. P. Ellis and G. E. Poliner, "Identifying 'cover songs' with chroma features and dynamic programming beat tracking," in *2007 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 4, 2007, pp. IV–1429–IV–1432.

[3] S. Dieleman, A. van den Oord, and K. Simonyan, "The challenge of realistic music generation: modelling raw audio at scale," *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[4] S. Lattner, M. Grachten, and G. Widmer, "Imposing higher-level structure in polyphonic music generation using convolutional restricted boltzmann machines and constraints," *Journal of Creative Music Systems*, vol. 2, pp. 1–31, 2018.

[5] D. P. Ellis, "Beat tracking by dynamic programming," *Journal of New Music Research*, vol. 36, no. 1, pp. 51–60, 2007.

[6] S. Dixon, "Evaluation of the audio beat tracking system beatroot," *Journal of New Music Research*, vol. 36, no. 1, pp. 39–50, 2007.

[7] E. MatthewDavies and S. Böck, "Temporal convolutional networks for musical audio beat tracking," in *2019 27th European Signal Processing Conference (EUSIPCO)*. IEEE, 2019, pp. 1–5.

[8] F. Krebs, A. Holzapfel, A. T. Cemgil, and G. Widmer, "Inferring metrical structure in music using particle filters," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 5, pp. 817–827, 2015.

[9] S. Durand, J. P. Bello, B. David, and G. Richard, "Robust downbeat tracking using an ensemble of convolutional networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 1, pp. 76–89, 2016.

[10] M. Fuentes, B. McFee, H. C. Crayencour, S. Essid, and J. P. Bello, "A music structure informed downbeat tracking system using skip-chain conditional random fields and deep learning," in *2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 481–485.

[11] F. Gouyon and S. Dixon, "A review of automatic rhythm description systems," *Computer music journal*, vol. 29, no. 1, pp. 34–54, 2005.

[12] T. Grill and J. Schlüter, "Music boundary detection using neural networks on combined features and two-level annotations." in *Proceedings of the 16th International Society for Music Information Retrieval Conference (ISMIR)*, 2015, pp. 531–537.

[13] B. McFee and D. Ellis, "Analyzing song structure with spectral clustering." in *Proceedings of the 15th International Society for Music Information Retrieval Conference (ISMIR)*. Citeseer, 2014, pp. 405–410.

[14] J. Salamon, O. Nieto, and N. J. Bryan, "Deep embeddings and section fusion improve music segmentation," in *Proceedings of the 22th International Society for Music Information Retrieval Conference (ISMIR)*, 2021.

[15] M. C. McCallum, "Unsupervised learning of deep features for music segmentation," in *2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 346–350.

[16] F. Lerdahl and R. S. Jackendoff, *A Generative Theory of Tonal Music, reissue, with a new preface*. MIT press, 1996.

[17] J. M. Robins, "Phrase structure, hypermeter, and closure in popular music," Ph.D. dissertation, The Florida State University, 2017.

[18] J. Wojcik and B. Kostek, "Representations of music in ranking rhythmic hypotheses," in *Advances in Music Information Retrieval*. Springer, 2010, pp. 39–64.

[19] D. Temperley, "Hypermetrical transitions," *Music Theory Spectrum*, vol. 30, no. 2, pp. 305–325, 2008.

[20] B. Kostek, J. Wojcik, and P. Szczuko, "Searching for metric structure of musical files," in *International Conference on Rough Sets and Intelligent Systems Paradigms*. Springer, 2007, pp. 774–783.

[21] Y.-C. Chuang and L. Su, "Beat and downbeat tracking of symbolic music data using deep recurrent neural networks," in *2020 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*. IEEE, 2020, pp. 346–352.

[22] B. Meudic, "Automatic meter extraction from MIDI files," in *Journées d'Informatique Musicale*, Marseille, France, Jun. 2002, pp. 1–1.

[23] J. C. Brown, "Determination of the meter of musical scores by autocorrelation," *The Journal of the Acoustical Society of America*, vol. 94, no. 4, pp. 1953–1957, 1993.

[24] D. Eck and N. Casagrande, "Finding meter in music using an autocorrelation phase matrix and shannon entropy." in *Proceedings of the 6th International Conference on Music Information Retrieval (ISMIR)*, 2005, pp. 504–509.

[25] W. B. De Haas and A. Volk, "Meter detection in symbolic music using inner metric analysis," in *Proceedings of the 17th International Society for Music Information Retrieval Conference (ISMIR)*, 2016, p. 441.

[26] C. J. Tralie and B. McFee, "Enhanced hierarchical music structure annotations via feature level similarity fusion," in *2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 201–205.

[27] K. Ullrich, J. Schlüter, and T. Grill, "Boundary detection in music structure analysis using convolutional neural networks." in *Proceedings of the 15th International Society for Music Information Retrieval Conference (ISMIR)*, 2014, pp. 417–422.

[28] R. B. Dannenberg and M. Goto, "Music structure analysis from acoustic signals," in *Handbook of signal processing in acoustics*. Springer, 2008, pp. 305–331.

[29] J. Paulus, M. Müller, and A. Klapuri, "State of the art report: Audio-based music structure analysis." in *Proceedings of the 11st International Society for Music Information Retrieval Conference (ISMIR)*. Utrecht, 2010, pp. 625–636.

[30] S. Werf and P. Hendriks, "A constraint-based approach to grouping in language and music," in *Proceedings of the Conference on Interdisciplinary Musicology*. Department of Musicology, University of Graz, 2004.

[31] S. Dai, H. Zhang, and R. B. Dannenberg, "Automatic analysis and influence of hierarchical structure on melody, rhythm and harmony in popular music," *arXiv preprint arXiv:2010.07518*, 2020.

[32] J. A. Jones, B. O. Miller, and D. L. Scarborough, "A rule-based expert system for music perception," *Behavior Research Methods, Instruments, & Computers*, vol. 20, no. 2, pp. 255–262, 1988.

[33] D.-J. Povel, "A theoretical framework for rhythm perception," *Psychological research*, vol. 45, no. 4, pp. 315–337, 1984.

[34] D. F. Rosenthal, "Machine rhythm–computer emulation of human rhythm perception," Ph.D. dissertation, Massachusetts Institute of Technology, 1992.

[35] D. Temperley and D. Sleator, "Modeling meter and harmony: A preference-rule approach," *Computer Music Journal*, vol. 23, no. 1, pp. 10–27, 1999.

[36] M. Hamanaka, K. Hirata, and S. Tojo, "Implementing "a generative theory of tonal music," *Journal of New Music Research*, vol. 35, no. 4, pp. 249–277, 2006.

[37] D. Temperley, "A unified probabilistic model for polyphonic music analysis," *Journal of New Music Research*, vol. 38, no. 1, pp. 3–18, 2009.

[38] M. Hamanaka, K. Hirata, and S. Tojo, "deepgttm-i&ii: Local boundary and metrical structure analyzer based on deep learning technique," in *International Symposium on Computer Music Multidisciplinary Research*. Springer, 2016, pp. 3–21.

[39] ——, "deepgttm-iii: simultaneous learning of grouping and metrical structures," in *the 13th International Symposium on Computer Music Multidisciplinary Research (CMMR2017)*, 2017, pp. 161–172.

[40] M. Rohrmeier, "Towards a formalization of musical rhythm," in *Proceedings of the 19th International Society for Music Information Retrieval Conference (ISMIR)*, 2020, pp. 621–629.

[41] S. Böck, M. E. Davies, and P. Knees, "Multi-task learning of tempo and beat: Learning one to improve the other." in *Proceedings of the 20th International Society for Music Information Retrieval Conference (ISMIR)*, 2019, pp. 486–493.

[42] S. Böck and M. E. Davies, "Deconstruct, analyse, reconstruct: How to improve tempo, beat, and downbeat estimation," in *Proceedings of the 21st International Society for Music Information Retrieval Conference (ISMIR)*, 2020, pp. 12–16.

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

[44] S. Durand and S. Essid, "Downbeat detection with conditional random fields and deep learned features." in *Proceedings of the 17th International Society for Music Information Retrieval Conference (ISMIR)*, 2016, pp. 386–392.

[45] H. V. Koops, W. B. de Haas, J. Bransen, and A. Volk, "Chord label personalization through deep learning of integrated harmonic interval-based representations," *arXiv preprint arXiv:1706.09552*, 2017.

[46] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[47] S. Tojo, K. Hirata, and M. Hamanaka, "Computational reconstruction of cognitive music theory," *New Generation Computing*, vol. 31, no. 2, pp. 89–113, 2013.

[48] J. Serrà, M. Müller, P. Grosche, and J. L. Arcos, "Unsupervised detection of music boundaries by time series structure features," in *AAAI*, 2012.

[49] Z. Wang, K. Chen, J. Jiang, Y. Zhang, M. Xu, S. Dai, and G. Xia, "POP909: A pop-song dataset for music arrangement generation," in *Proceedings of the 21st International Society for Music Information Retrieval Conference (ISMIR)*. Montreal, Canada: ISMIR, Oct. 2020, pp. 38–45.

[50] C. Raffel, "Learning-based methods for comparing sequences, with applications to audio-to-midi alignment and matching," Ph.D. dissertation, 2016.