Downbeat Tracking for Western Classical Music Recordings: A Case Study for Beethoven Piano Sonatas

Ching-Yu Chiu Johannes Zeitler Vlora Arifi-Müller Meinard Müller

International Audio Laboratories Erlangen, Germany

{*ching-yu.chiu, meinard.mueller*}*@audiolabs-erlangen.de*

Abstract

Tracking beats and downbeats are fundamental skills that enable humans to comprehend music and engage with it. While both beats and downbeats exhibit periodicity over time, estimating downbeats demands a deeper understanding of musical aspects, such as onsets, beats, melodies, phrases, and thus requires a larger musical context. To assess the efficacy of models in learning downbeats, it is crucial to utilize datasets of different musical styles that encompass varying degrees of complexity, tempo changes, and expressivity. However, due to the scarcity of high-quality annotated datasets of expressive classical music, the performance and behavior of state-of-the-art downbeat tracking models is largely unexplored in this context. In this study, we conduct a comprehensive performance analysis of existing downbeat tracking models using a carefully curated dataset of Beethoven Piano Sonatas with downbeat annotations, comprising pieces and performances of various levels of expressivity. In particular, we use context-sensitive and metric-level-sensitive evaluation measures to better understand the models' benefits and limitations. Furthermore, we explore the impact of training data, categorize sources of errors, and suggest potential directions for future research in this area.

Introduction

Beats are typically referred to as the time positions humans would tap along with when listening to music. Downbeats are the first beat of each measure. While 'downbeat tracking' is often used for popular music, 'measure detection/estimation' is often used for Western classical music where music scores are available. Although both beats and downbeats usually go along with note onsets, the determination of downbeats requires a larger musical context involving various musical elements/aspects (e.g., chords, keys, phrases), making it a more intricate task [1, 2].

Despite the success and dominance of methods based on deep learning (DL) in the field of beat and downbeat tracking [3–5], their performance and behavior for expressive classical music often fall short of expectations or remain largely unexplored. It is therefore our goal in this study to evaluate and analyze existing state-of-the-art (SOTA) models using a carefully curated dataset of Beethoven Piano Sonatas (BPSD) [6] with reference downbeat annotations. Figure 1 shows the components of an existing downbeat tracking system of the example of Beethoven's 6th piano sonata, played by Wilhelm Kempff. Given an audio input feature representation (Figure 1a), existing downbeat tracking systems produce downbeat activation functions (Figure 1b) indicating the probability of each time frame to be a downbeat. The activation functions are then post-processed by a model-based method (e.g., dynamic programming [7]) and converted into downbeat estimates (Figure 1c). With the corresponding reference downbeats and musical score, sources of errors can be further analyzed.

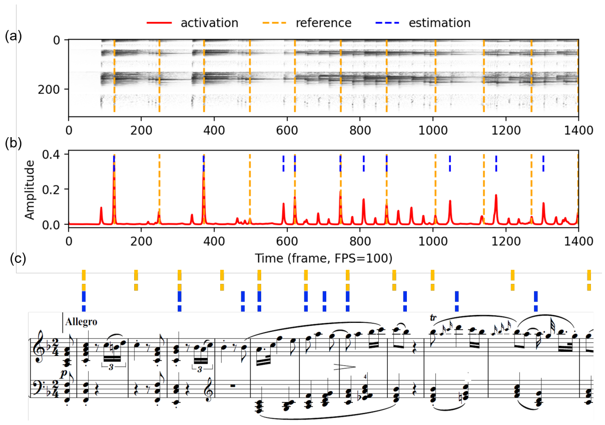


Figure 1: Components of a downbeat tracking system. **(a)** Audio feature. **(b)** Downbeat activation function. **(c)** The reference/estimated downbeats and the music score.

In this study, we conduct experiments using SOTA activation functions in combination with different post-processors. Experiment results indicate that one of the main challenges of downbeat tracking may come from the potential ambiguity of the definition of downbeats in the audio domain. Even though the definition of downbeats is quite clear in music scores (i.e., the symbolic domain), it becomes less strict in the audio domain as one may consider several plausible interpretations as reasonable when the music score is unknown. As humans perceive downbeats via jointly processing several aspects of music, the concept of downbeats is ambiguously related to multiple levels of musical elements. This ambiguity of downbeat definition manifests in the failure of both activation functions and post-processing of SOTA methods. Based on a coherent multi-version dataset of Beethoven's piano sonatas (BPSD) [6], we will discuss and further explore some of these issues in more detail.

ID	Performer	Year	Duration
AS35	Artur Schnabel	1935	03:33:35
FG58	Friedrich Gulda	1958	03:34:00
FJ62	Fritz Jank	1962	03:41:26
WK64	Wilhelm Kempff	1964	03:45:31
FG67	Friedrich Gulda	1967	03:25:02
VA81	Vladimir Ashkenazy	1981	03:46:27
DB84	Daniel Barenboim	1984	03:58:37
JJ90	Jeno Jando	1990	03:39:14
AB96	Alfred Brendel	1996	03:52:28
MB97	Malcolm Bilson et al.	1997	03:46:08
MC22	Muriel Chemin	2022	04:05:11
		Total	41:07:45

Table 1: Overview of audio versions in the BPSD. The versions with identifiers AS35, FG58, FJ62, and WK64 are in the public domain and are freely accessible within the BPSD. Durations given in hh:mm:ss.

Dataset

We conduct our experiments on classical *piano* music, as it provides well-defined onsets as opposed to, e.g., string or choir music, where onsets are less well-defined and therefore additional challenges beyond beat or downbeat tracking arise. To this end, we choose the Beethoven Piano Sonatas Dataset (BPSD) as our evaluation corpus, which comprises the first movements of the 32 piano sonatas by Ludwig van Beethoven. Expanding and further developing the traditional sonata form, Beethoven's piano sonatas rank among the most pivotal works in the history of Western classical music and heavily influenced later composers. Expressive and dramatic recordings of the sonatas are available in a multitude of different interpretations. The BPSD includes eleven complete audio recordings of the 32 sonatas (see Table 1), encompassing performances on historic instruments, live performances, vintage recordings with low audio quality, and modern studio recordings. Measure positions in the BPSD were annotated manually for the recordings by Wilhelm Kempff (WK64) and automatically transferred to all other audio versions using high-resolution audio–audio synchronization techniques.

Methods

Table 2 (top) describes two sources of the open-source implementations for computing DL-based activation functions adopted in this work. We use RNNDownBeatProcessor from madmom [3, 4], which is based on bidirectional long short-term memory networks (BLSTMs), noted as MAD-*. Since 2022, several transformer-based beat/downbeat trackers have been proposed [5,8]. In our experiments, we adopted an open-source model, beat-transformer (BTF-*) [5] in this work. All activation functions are real-valued with values between 0 and 1.

Table 2 (bottom) describes the main post-processors adopted in this work: a dynamic programming-based post-processor [7] (DP), and a simple peak picker (PP) from Scipy[1]. The DP-based method requires a reference global tempo to find a tradeoff between tempo consistency and intensity of the activation functions. The PP method makes no assumption for tempo and picks all activation peaks that fulfill the specified critera of a peak (see Table 2). In this work, we use the DP implemented in [10] with the track-wise mean inter-measure interval (IMI) calculated from downbeat annotations to derive the global tempo information.

Activation Functions	
MAD-B	Beat activation function of madmom.
MAD-D	Downbeat activation function of madmom.
BTF-B	Beat activation function of beat transformer
BTF-D	Downbeat activation function of beat transformer
Post- Processors	
PP	A simple peak picker from Scipy. $\text{threshold} = 0.1, \text{distance} = 7, \text{prominence} = 0.1$
DP	Dynamic programming-based post-processor. Track-wise mean inter-measure interval (IMI) as reference information.

Table 2: Activation functions and post-processors.

Besides the conventional evaluation metrics of F1-score (F), precision (P), and recall (R), we also include one context-sensitive evaluation metric, referred to as L-correctness [11]. Instead of considering whether a single downbeat is correctly matched, the L-correctness metric requires at least L consecutive downbeats being matched correctly. By increasing the L value, one can evaluate the downbeat trackers in a stricter manner, inspecting the efficacy of the models in handling broader musical contexts. For the following experiments, we set $L = 2$ and report the F1-score of L-correctness[2]. A tolerance window size of ±70 ms is used for all metrics.

Methods	P	R	F	L2F
BTF-D_PP	0.405	0.635	**0.478**	0.208
BTF-D_DP	**0.467**	0.474	**0.470**	**0.407**
MAD-D_PP	**0.477**	0.485	0.460	0.229
MAD-D_DP	0.417	0.424	0.420	**0.352**
MAD-B_PP	0.300	**0.790**	0.422	0.061
BTF-B_PP	0.272	**0.843**	0.402	0.037

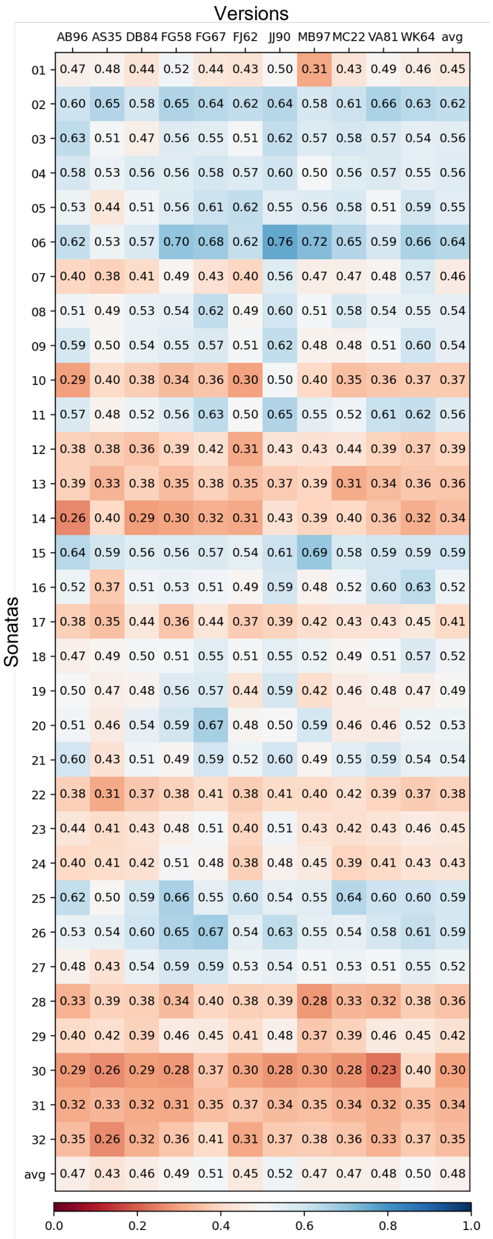
Table 3: Downbeat tracking performance.

Experiments: Overall Results

Table 3 shows the overall downbeat tracking performance of the adopted models for the BPSD. Results are sorted based on the F1-scores. In general, it can be observed that the existing methods do not work well for expressive classical music in the BPSD. For example, even the best-performing model BTF-D_PP achieves an F1-score of only 0.478. As the PP method does not make any assumption regarding the tempo and generally picks all peaks as downbeats, it is not surprising to have low precision val-

[1] Note that, similar to the findings in [9], the widely used HMM-based post processor [3, 4] completely fails for BPSD, regardless of the settings of the parameter, transition_lambda. We therefore excluded the HMM in this study.

[2] Readers may refer to [11] for other detailed aspects of L-correctness.

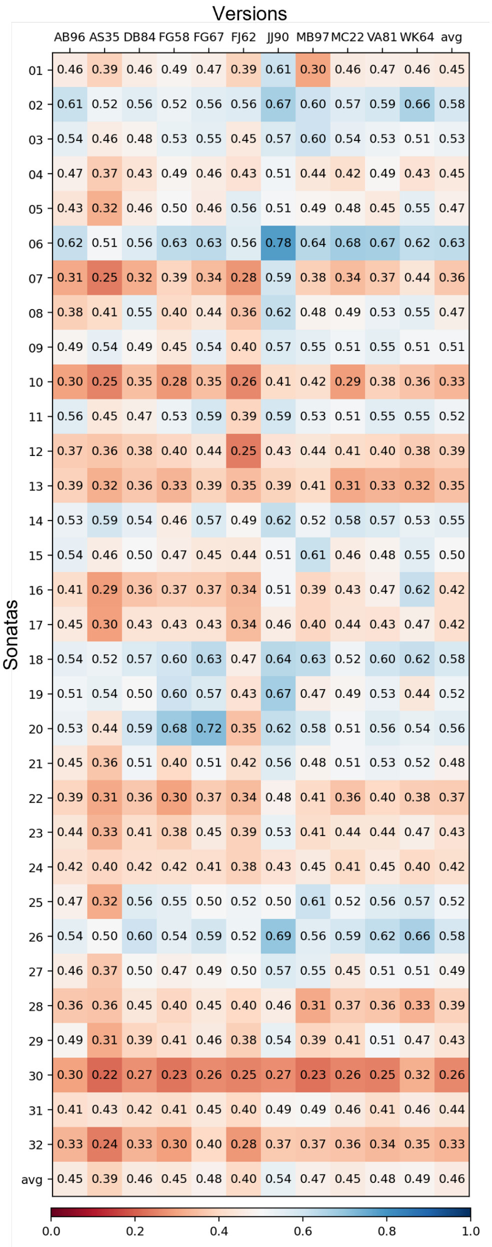


Sonatas	AB96	AS35	DB84	FG58	FG67	FJ62	JJ90	MB97	MC22	VA81	WK64	avg
01	0.47	0.48	0.44	0.52	0.44	0.43	0.50	0.31	0.43	0.49	0.46	0.45
02	0.60	0.65	0.58	0.65	0.64	0.62	0.64	0.58	0.61	0.66	0.63	0.62
03	0.63	0.51	0.47	0.56	0.55	0.51	0.62	0.57	0.58	0.57	0.54	0.56
04	0.58	0.53	0.56	0.56	0.58	0.57	0.60	0.50	0.56	0.57	0.55	0.56
05	0.53	0.44	0.51	0.56	0.61	0.62	0.55	0.56	0.58	0.51	0.59	0.55
06	0.62	0.53	0.57	0.70	0.68	0.62	0.76	0.72	0.65	0.59	0.66	0.64
07	0.40	0.38	0.41	0.49	0.43	0.40	0.56	0.47	0.47	0.48	0.57	0.46
08	0.51	0.49	0.53	0.54	0.62	0.49	0.60	0.51	0.58	0.54	0.55	0.54
09	0.59	0.50	0.54	0.55	0.57	0.51	0.62	0.48	0.48	0.51	0.60	0.54
10	0.29	0.40	0.38	0.34	0.36	0.30	0.50	0.40	0.35	0.36	0.37	0.37
11	0.57	0.48	0.52	0.56	0.63	0.50	0.65	0.55	0.52	0.61	0.62	0.56
12	0.38	0.38	0.36	0.39	0.42	0.31	0.43	0.43	0.44	0.39	0.37	0.39
13	0.39	0.33	0.38	0.35	0.38	0.35	0.37	0.39	0.31	0.34	0.36	0.36
14	0.26	0.40	0.29	0.30	0.32	0.31	0.43	0.39	0.40	0.36	0.32	0.34
15	0.64	0.59	0.56	0.56	0.57	0.54	0.61	0.69	0.58	0.59	0.59	0.59
16	0.52	0.37	0.51	0.53	0.51	0.49	0.59	0.48	0.52	0.60	0.63	0.52
17	0.38	0.35	0.44	0.36	0.44	0.37	0.39	0.42	0.43	0.43	0.45	0.41
18	0.47	0.49	0.50	0.51	0.55	0.51	0.55	0.52	0.49	0.51	0.57	0.52
19	0.50	0.47	0.48	0.56	0.57	0.44	0.59	0.42	0.46	0.48	0.47	0.49
20	0.51	0.46	0.54	0.59	0.67	0.48	0.50	0.59	0.46	0.46	0.52	0.53
21	0.60	0.43	0.51	0.49	0.59	0.52	0.60	0.49	0.55	0.59	0.54	0.54
22	0.38	0.31	0.37	0.38	0.41	0.38	0.41	0.40	0.42	0.39	0.37	0.38
23	0.44	0.41	0.43	0.48	0.51	0.40	0.51	0.43	0.42	0.43	0.46	0.45
24	0.40	0.41	0.42	0.51	0.48	0.38	0.48	0.45	0.39	0.41	0.43	0.43
25	0.62	0.50	0.59	0.66	0.55	0.60	0.54	0.55	0.64	0.60	0.60	0.59
26	0.53	0.54	0.60	0.65	0.67	0.54	0.63	0.55	0.54	0.58	0.61	0.59
27	0.48	0.43	0.54	0.59	0.59	0.53	0.54	0.51	0.53	0.51	0.55	0.52
28	0.33	0.39	0.38	0.34	0.40	0.38	0.39	0.28	0.33	0.32	0.38	0.36
29	0.40	0.42	0.39	0.46	0.45	0.41	0.48	0.37	0.39	0.46	0.45	0.42
30	0.29	0.26	0.29	0.28	0.37	0.30	0.28	0.30	0.28	0.23	0.40	0.30
31	0.32	0.33	0.32	0.31	0.35	0.37	0.34	0.35	0.34	0.32	0.35	0.34
32	0.35	0.26	0.32	0.36	0.41	0.31	0.37	0.38	0.36	0.33	0.37	0.35
avg	0.47	0.43	0.46	0.49	0.51	0.45	0.52	0.47	0.47	0.48	0.50	0.48

Figure 2: Downbeat tracking performance of BTF-D_PP.

ues for BTF-D_PP ($P = 0.405$) and MAD-D_PP ($P = 0.477$). Based on the similar rationale, the low recall values of BTF-D_PP ($R = 0.635$) and MAD-D_PP ($R = 0.485$) indicate that there are many missing or weak peaks[3] at downbeat positions of BTF-D and MAD-D. When considering the context-sensitive L-correctness, the evaluation metric yields even lower values. For example, one obtains $L2F = 0.208$ for BTF-D_PP, indicating the limited number of consecutive downbeats being detected together.

The results of BTF-D_DP and MAD-D_DP demonstrate a different behavior when a global IMI is provided to the downbeat tracker to enforce a stable tempo. Specifically, the stable tempo assumption of DP improves the L2F measure (e.g., from 0.208 to 0.407 for BTF-D, and from



Sonatas	AB96	AS35	DB84	FG58	FG67	FJ62	JJ90	MB97	MC22	VA81	WK64	avg
01	0.46	0.39	0.46	0.49	0.47	0.39	0.61	0.30	0.46	0.47	0.46	0.45
02	0.61	0.52	0.56	0.52	0.56	0.56	0.67	0.60	0.57	0.59	0.66	0.58
03	0.54	0.46	0.48	0.53	0.55	0.45	0.57	0.60	0.54	0.53	0.51	0.53
04	0.47	0.37	0.43	0.49	0.46	0.43	0.51	0.44	0.42	0.49	0.43	0.45
05	0.43	0.32	0.46	0.50	0.46	0.56	0.51	0.49	0.48	0.45	0.55	0.47
06	0.62	0.51	0.56	0.63	0.63	0.56	0.78	0.64	0.68	0.67	0.62	0.63
07	0.31	0.25	0.32	0.39	0.34	0.28	0.59	0.38	0.34	0.37	0.44	0.36
08	0.38	0.41	0.55	0.40	0.44	0.36	0.62	0.48	0.49	0.53	0.55	0.47
09	0.49	0.54	0.49	0.45	0.54	0.40	0.57	0.55	0.51	0.55	0.51	0.51
10	0.30	0.25	0.35	0.28	0.35	0.26	0.41	0.42	0.29	0.38	0.36	0.33
11	0.56	0.45	0.47	0.53	0.59	0.39	0.59	0.53	0.51	0.55	0.55	0.52
12	0.37	0.36	0.38	0.40	0.44	0.25	0.43	0.44	0.41	0.40	0.38	0.39
13	0.39	0.32	0.36	0.33	0.39	0.35	0.39	0.41	0.31	0.33	0.32	0.35
14	0.53	0.59	0.54	0.46	0.57	0.49	0.62	0.52	0.58	0.57	0.53	0.55
15	0.54	0.46	0.50	0.47	0.45	0.44	0.51	0.61	0.46	0.48	0.55	0.50
16	0.41	0.29	0.36	0.37	0.37	0.34	0.51	0.39	0.43	0.47	0.62	0.42
17	0.45	0.30	0.43	0.43	0.43	0.34	0.46	0.40	0.44	0.43	0.47	0.42
18	0.54	0.52	0.57	0.60	0.63	0.47	0.64	0.63	0.52	0.60	0.62	0.58
19	0.51	0.54	0.50	0.60	0.57	0.43	0.67	0.47	0.49	0.53	0.44	0.52
20	0.53	0.44	0.59	0.68	0.72	0.35	0.62	0.58	0.51	0.56	0.54	0.56
21	0.45	0.36	0.51	0.40	0.51	0.42	0.56	0.48	0.51	0.53	0.52	0.48
22	0.39	0.31	0.36	0.30	0.37	0.34	0.48	0.41	0.36	0.40	0.38	0.37
23	0.44	0.33	0.41	0.38	0.45	0.39	0.53	0.41	0.44	0.44	0.47	0.43
24	0.42	0.40	0.42	0.42	0.41	0.38	0.43	0.45	0.41	0.45	0.40	0.42
25	0.47	0.32	0.56	0.55	0.50	0.52	0.50	0.61	0.52	0.56	0.57	0.52
26	0.54	0.50	0.60	0.54	0.59	0.52	0.69	0.56	0.59	0.62	0.66	0.58
27	0.46	0.37	0.50	0.47	0.49	0.50	0.57	0.55	0.45	0.51	0.51	0.49
28	0.36	0.36	0.45	0.40	0.45	0.40	0.46	0.31	0.37	0.36	0.33	0.39
29	0.49	0.31	0.39	0.41	0.46	0.38	0.54	0.39	0.41	0.51	0.47	0.43
30	0.30	0.22	0.27	0.23	0.26	0.25	0.27	0.23	0.26	0.25	0.32	0.26
31	0.41	0.43	0.42	0.41	0.45	0.40	0.49	0.49	0.46	0.41	0.46	0.44
32	0.33	0.24	0.33	0.30	0.40	0.28	0.37	0.37	0.36	0.34	0.35	0.33
avg	0.45	0.39	0.46	0.45	0.48	0.40	0.54	0.47	0.45	0.48	0.49	0.46

Figure 3: Downbeat tracking performance of MAD-D_PP.

0.229 to 0.352 for MAD-D) at the cost of a lower recall (BTF-D: from 0.635 to 0.474, and from 0.485 to 0.424 for MAD-D).

We further evaluate the beat estimations of MAD-B_PP and BTF-B_PP using downbeat reference annotations (see Table 3 bottom). While it is not surprising that the beat trackers get much lower precision values ($P = 0.272$ for BTF-B_PP) and L2F values (0.037) when evaluated as downbeats, it is worth noting that the recall values are much higher than the above downbeat trackers ($R = 0.843$ for BTF-B_PP compared to $R = 0.635$ for BTF-D_PP). This further indicates the following two issues. First, the low recall values of BTF-D_PP and MAD-D_PP imply that existing training mechanisms do not help the models to really learn the idea of downbeats. Second, the downbeat tracking performance may be improved by incorporating an explicit mechanism to select downbeats from the beat

[3] Note that the peak threshold of PP is set to 0.1 to require basic peak height. However, we found the downbeat activation peaks for BPSD are often below 0.1 and therefore lead to the low recall.

activation peaks (of BTF-B and MAD-B).

Experiments: Track-Wise Results

Figure 2 and 3 show the track-wise F1-scores of BTF-D_PP and MAD-D_PP, providing both sonata view and version view for us to gain deeper understanding of the downbeat tracking results. For better visibility of trends across sonatas and versions, we color-code the F1-scores using blue for ($F > 0.5$) and red for ($F < 0.5$). Similar to our findings in the overall results, there are both consistent trends and contradictory behaviors between BTF-D and MAD-D. For example, from the similar distribution of blue and red cells, we can see that, both models perform relatively good for sonatas 02, 06, and 26 and perform worse for sonatas 01, 07, 10, 12, 13, 22-24, and 28-32. By sonifying and visualizing the results, we found that this is because existing models generally produce activation peaks at onsets that are emphasized by the pianists. Therefore sonatas with specific musical properties tend to be more challenging for these models. Looking at specific performances or versions, both models perform relatively good for JJ90 and relatively poor for AS35. This is mainly due to the audio quality. While the JJ90 consists of modern recordings of high audio quality, the AS35 consists of old recordings of poor quality. We also observe some inconsistent behaviors between BTF-D and MAD-D. While sonata 14 is easier for MAD-D (blue), it is harder for BTF-D (red). From the different levels of contrast across versions, we can also see that MAD-D seems to be more sensitive to different interpretations of pianists. For example, for sonata 25 the F1-scores range from 0.32 to 0.61 for MAD-D while stay within 0.50 to 0.66 for BTF-D. These inconsistent behaviors of the models indicate that they may learn and rely on different patterns for downbeat tracking.

Conclusion and Future Directions

Based on the above experiments and discussions, one can see that existing models suffer from issues including confusion of tasks (i.e., beats vs. downbeats), low confidence (i.e., weak/missing activation peaks at downbeat positions), and low control of learned patterns (i.e., contradictory behaviors between BTF-D and MAD-D). This all goes along with the ambiguity of the definition of downbeats. To tackle the challenging task of downbeat tracking for expressive classical music, the relevant information and mechanisms we humans utilize (e.g., perception of chord progression and musical structures) when doing downbeat tracking need to be more explicitly incorporated into the feature representations, model architectures, objective functions, and model evaluation. Moreover, multi-version datasets such as BPSD or a similar cross-version dataset for five Chopin Mazurkas Dataset [12] providing various aspects of musical annotations will also play important roles to understand and improve the models.

References

[1] H. Papadopoulos and G. Peeters, "Joint estimation of chords and downbeats from an audio signal," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 1, pp. 138–152, 2011.

[2] C. Weiß, V. Arifi-Müller, T. Prätzlich, R. Kleinertz, and M. Müller, "Analyzing measure annotations for Western classical music recordings," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, New York, USA, 2016, pp. 517–523.

[3] S. Böck, F. Korzeniowski, J. Schlüter, F. Krebs, and G. Widmer, "madmom: A new Python audio and music signal processing library," in *Proceedings of the ACM International Conference on Multimedia (ACM-MM)*, Amsterdam, The Netherlands, 2016, pp. 1174–1178.

[4] S. Böck, F. Krebs, and G. Widmer, "Joint beat and downbeat tracking with recurrent neural networks," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, New York City, New York, USA, 2016, pp. 255–261.

[5] J. Zhao, G. Xia, and Y. Wang, "Beat transformer: Demixed beat and downbeat tracking with dilated self-attention," in *ISMIR*, 2022, pp. 169–177.

[6] J. Zeitler, C. Weiß, V. Arifi-Müller, and M. Müller, "BPSD: A coherent multi-version dataset for analyzing the first movements of beethoven's piano sonatas," *Transactions of the International Society for Music Information Retrieval (TISMIR)*, submitted 2024.

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

[8] Y. Hung, J. Wang, X. Song, W. T. Lu, and M. Won, "Modeling beats and downbeats with a time-frequency transformer," in *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP*, 2022, pp. 401–405.

[9] C. Chiu, M. Müller, M. E. P. Davies, A. W. Su, and Y. Yang, "Local periodicity-based beat tracking for expressive classical piano music," *IEEE ACM Trans. Audio Speech Lang. Process.*, vol. 31, pp. 2824–2835, 2023.

[10] M. Müller and F. Zalkow, "libfmp: A Python package for fundamentals of music processing," *Journal of Open Source Software (JOSS)*, vol. 6, no. 63, pp. 3326:1–5, 2021.

[11] P. Grosche and M. Müller, "Extracting predominant local pulse information from music recordings," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 6, pp. 1688–1701, 2011.

[12] P. Grosche, M. Müller, and C. S. Sapp, "What makes beat tracking difficult? A case study on Chopin Mazurkas," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Utrecht, The Netherlands, 2010, pp. 649–654.