MATCHMAKER: AN OPEN-SOURCE LIBRARY FOR REAL-TIME PIANO SCORE FOLLOWING AND SYSTEMATIC EVALUATION

Jiyun Park[1*] **Carlos Cancino-Chacón**[2*]
Suhit Chiruthapudi[2] **Juhan Nam**[1]
[1] Graduate School of Culture Technology, KAIST, South Korea
[2] Institute of Computational Perception, Johannes Kepler University Linz, Austria
{june,juhan.nam}@kaist.ac.kr,
{carlos.cancino_chacon,suhit.chiruthapudi}@jku.at

ABSTRACT

Real-time music alignment, also known as *score following*, is a fundamental MIR task with a long history and is essential for many interactive applications. Despite its importance, there has not been a unified open framework for comparing models, largely due to the inherent complexity of real-time processing and the language- or system-dependent implementations. In addition, low compatibility with the existing MIR environment has made it difficult to develop benchmarks using large datasets available in recent years. While new studies based on established methods (e.g., dynamic programming, probabilistic models) have emerged, most evaluations compare models only within the same family or on small sets of test data. This paper introduces *Matchmaker*, an open-source Python library for real-time music alignment that is easy to use and compatible with modern MIR libraries. Using this, we systematically compare methods along two dimensions: music representations and alignment methods. We evaluated our approach on a large test set of solo piano music from the (n)ASAP, Batik, and Vienna4x22 datasets with a comprehensive set of metrics to ensure robust assessment. Our work aims to establish a benchmark framework for score-following research while providing a practical tool that developers can easily integrate into their applications.

1. INTRODUCTION

Real-time music alignment, also known as *score following*, is the task of aligning performance data to the corresponding position in the musical score in real-time. Ever since it was first introduced independently by Roger Dannenberg [1] and Barry Vercoe [2] over 40 years ago, music alignment has become one of the fundamental MIR tasks. Score following is a necessary component of many interactive applications (e.g., automatic accompaniment systems [3–6], automatic page turning [7, 8], lyrics alignment or tracking singing voice [9–11], audiovisual/multimodal [6, 12] and visualizations [13]. Music alignment began as real-time score following [1, 2, 14–17] but, by the mid-90s, had diverged into online and offline methods (see, e.g., early offline work by Desain et al. [18]).

From its early use on monophonic sources like voice [17] and wind instruments, score following has grown to support polyphonic instruments such as piano, ensemble, and even full orchestral performances [17, 19–21]. Research has also expanded across input modalities of the performance, with systems operating on audio or MIDI, and score representations including string format, symbolic score, and sheet image [22].

The score following challenge [23] in MIREX laid the foundation to formalize the evaluation framework, introducing important metrics that include considerations in real-time. However, many subsequent studies have been developed in different environments—ranging from system-dependent [24, 25] to language-dependent [26, 27] implementations—often tailored to specific use cases and without publicly shared source code. As a result, implementations became fragmented across platforms, making it difficult to extend, reproduce, or compare methods in a unified setting. This has hindered the development of a unified evaluation framework and comparison over methods or features on shared datasets remain rare, limiting the generalizability and reproducibility.

In this paper, we address these challenges by proposing a unified, open framework for the evaluation and benchmarking of real-time audio-based score following. Considering public datasets that offer a range of difficulty levels, multiple renditions, and precise beat-level annotations, we base our evaluation on three representative piano performance datasets. We implement this framework as an open-source Python package called *Matchmaker*, [1] that allows real-time execution of representative baselines of score following algorithms. In addition to benchmarking, it supports audio device input and has been validated in application contexts through a standalone demo system.

* Equal contribution.

© J. Park, C. Cancino-Chacón, S. Chiruthapudi and J. Nam. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** J. Park, C. Cancino-Chacón, S. Chiruthapudi and J. Nam, "Matchmaker: An Open-Source Library for Real-Time Piano Score Following and Systematic Evaluation", in *Proc. of the 26th Int. Society for Music Information Retrieval Conf.,* Daejeon, South Korea, 2025.

[1] https://github.com/pymatchmaker/matchmaker

2. A CONCEPTUAL FRAMEWORK FOR SCORE FOLLOWING

As a way to organize and compare the components of systems for score following, we follow the structure proposed by Müller [28]. This framework consists of three core components: (1) input music representations, (2) features, and (3) online alignment algorithms.

2.1 Music Representation

Score following aligns a fixed reference derived from musical scores with a time-evolving input from a performance. The score can take various symbolic formats (e.g., MIDI, MusicXML) or sheet images, and is typically converted into an intermediate representation such as synthesized audio or event sequences. The performance input may be given as either audio or MIDI, each with distinct representational and computational characteristics. Audio input is continuous and latency-sensitive, while MIDI is discrete and event-based. Instrumental factors also affect alignment design: polyphonic or discrete-pitch instruments (e.g., piano) differ from continuous-pitch sources (e.g., violin, voice). Multi-instrument recordings pose further challenges due to timbral overlap and source ambiguity.

2.2 Features

Chroma features are the most commonly used in music synchronization, with many variants for their computation [29–32]. Other works also use various spectral features such as constant-Q transforms (CQT) [27, 33], non-negative matrix factorization(NMF)-based [34] or spectral template [35] for improved polyphonic alignment. Beyond spectral representations, context-aware features such as onset-based feature [36] or beat-synchronous frames have been introduced to capture temporally salient events useful for alignment. Later work explored learned features, including feedforward mappings [27], semi-supervised decompositions like NMF, and more recent neural approaches [37]. While these offer richer contextual information, they often rely on fixed-length inputs and introduce latency, making real-time usage more challenging.

2.3 Alignment Algorithms

Two major families of alignment algorithms have been used in score following: dynamic programming and probabilistic models.

The dynamic programming approach, especially dynamic time warping (DTW), aligns two sequences by minimizing cumulative cost. Its online variant, On-Line Time Warping (OLTW) [38], enables causal alignment within a fixed-size of window. Variants include windowed [39], parallel [40], and constrained DTW [40, 41], as well as tempo-aware extensions [21, 42].

Probabilistic state-space models offer an alternative by treating alignment as latent state inference under uncertainty [24, 29, 43]. HMM-based systems model each note as a sequence of states (e.g., attack–steady–release), with extensions including semi-Markov [44], hybrid [19], and Bayesian variants [45]. Kalman filter models and switching state-space systems [46, 47] further incorporate tempo dynamics, while particle filters [12, 29] handle multimodal uncertainty in real time.

Other paradigms include early string-matching algorithms [1] and reinforcement learning-based approaches for multimodal or visual score alignment [48].

3. IMPLEMENTATION

3.1 Python Package Structure

Matchmaker is an open source Python package that implements representative real-time music alignment algorithms within a modular, extensible framework. Figure 1 illustrates the overview of the package and the whole pipeline. The current version of *Matchmaker* provides two types of algorithms: 1) online time warping, with two variants: `OLTWDixon`, based on the methods proposed in [38, 49], and `OLTWArzt`, based on [21, 50]; and 2) an HMM-based algorithm, similar to the one used in [3, 47]. A full description of the algorithms and their parameters can be found in the supplementary Appendix. [2]

Matchmaker supports two main usage scenarios: (1) live streaming mode using the audio device and (2) simulation mode, which processes a performance file as input. Figure 2 shows an example of running live streaming mode with the default setting. The `AudioStream` object handles the input stream by chunking the audio with overlapping windows to avoid padding artifacts. Both the synthesized score audio and the performance audio are passed to a `Processor` object that performs feature extraction. The extracted features are pushed into a queue and consumed by the `OnlineAlignment` object, which runs the alignment methods in real time. *Matchmaker* takes a musical score with all symbolic music formats (MusicXML, MIDI, MEI, etc.) available by *partitura*. [3] The returned output is the current position in the score, represented in beats as a musical unit according to the time signature in the piece. More detailed description and API documentation of the package are available here. [4]

3.2 Design and Implementation Details

We provide a simple and user-friendly interface to run the score following with minimal setup. As shown in Figure 2, users can instantiate a `Matchmaker` object with a score file and execute a run that iterates over the estimated score position for each step. To streamline real-time processing, the `AudioStream` class is implemented as a context manager that automatically handles stream initialization and teardown. Furthermore, the alignment process is designed as a generator, enabling users to receive score positions concurrently while the alignment is in progress.

[2] `https://pymatchmaker.github.io/ismir2025_supplementary_materials/`

[3] `https://github.com/CPJKU/partitura`

[4] `https://pymatchmaker.readthedocs.io/`

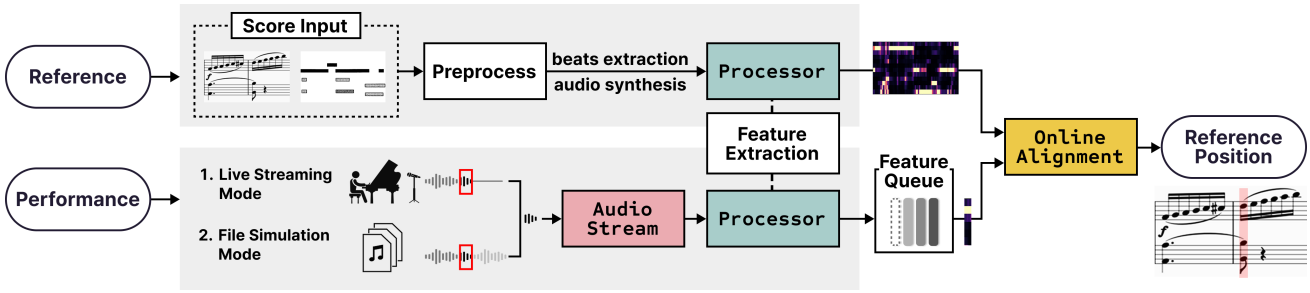

Figure 1. Overview of the score following package

```python
from matchmaker import Matchmaker

mm = Matchmaker(
    score_file="path/to/score.musicxml",
    input_type="audio",
)
for current_position in mm.run():
    print(current_position)
```

Figure 2. A code example for running the *Matchmaker* in a live streaming mode.

This design allows for efficient real-time integration without requiring users to manage multiple threads, buffers, or callbacks explicitly.

While the online mode uses a multi-threaded queue for asynchronous audio buffering, the simulation mode processes audio chunks in advance within a single-threaded setup. By decoupling real-time I/O concerns from core alignment evaluation, it is intended to avoid variability from Python version, OS-level threading, or queuing delays, ensuring a consistent and reproducible benchmarking environment. In addition, OLTWArzt is implemented in Cython [51] for efficiency, a superset of Python designed for C-like performance by incorporating C data types and optimizing the execution of Python code.

4. EXPERIMENTS

4.1 Datasets

We use three public piano performance datasets: (n)ASAP [52], Batik [53] and Vienna 4x22 [54], each of them offering complementary characteristics for benchmarking score following. (n)ASAP, a subset of the MAESTRO dataset including note-level score alignments, includes expressive performances of technically demanding solo piano pieces, offering high difficulty and stylistic diversity. We use only the pieces in the MAESTRO v2 test split. Vienna4x22 provides 22 distinct renditions for each of four relatively easy pieces, which is suitable to test robustness to interpretive variation. Batik dataset contains recordings of 12 Mozart sonatas by a single pianist with the longest average piece duration among the three datasets, enabling evaluation across long-form classical repertoire.

We use ground-truth beat-level annotations provided with the (n)ASAP dataset, and extract equivalent annotations for Batik and Vienna4x22 from the *.match* files [55], which contain note-wise score–performance alignments. In addition, we incorporate the difficulty levels of each piece based on G. Henle Publishers, [5] which provides a 1-to-9 grading scale. The pieces used in our experiments span levels 4 through 9, representing a diverse set of works above intermediate level. Table 1 provides the detailed statistics of the datasets.

Dataset	#Pieces	#Perf	#Beats	#Notes	Dur (h)	Difficulty
(n)ASAP	43	59	26,329	100,958	2.65	6.53
Batik	30	30	18,789	102,421	2.85	5.67
Vienna	4	88	13,728	43,656	2.24	4.88
Total	**77**	**177**	**58,846**	**247,035**	**7.74**	**6.11**

Table 1. Datasets used in the evaluation.

We only included performances in the experiment that recorded an MAE of less than 100 ms in the offline test, using the *synctoolbox* [6] with Chroma & DLNCO features. The evaluation was conducted on 184 performances across 93 pieces, totaling over 58,000 beats and 247,000 notes, with an overall duration of 7.74 hours of performances and a piece-wise average difficulty of 6.11.

4.2 Experiment Settings

We conducted all evaluations under simulation-based conditions to ensure reproducibility. Live testing was avoided due to variability introduced by room acoustics and hardware setup, which complicates fair comparison across systems. The accuracy tests were carried out on an Intel i9-9900K CPU (16 cores @ 3.6 GHz), Python 3.9, with a sample rate of 44.1 kHz and a frame rate of 30, chosen to balance latency and alignment accuracy. We tested chromagram, mel-spectrogram, constant-Q transform (CQT), mel-frequency cepstral coefficients (MFCCs) [56] and a simple STFT-based onset-sensitive representation similar to the one used in Dixon [38], which we name log-spectral energy (LSE). While results for all features were evaluated, we report detailed latency and accuracy metrics for the best-performing configuration of each model. To account for hardware variability, latency was measured in multiple setups: an Intel i9-9900K, an Apple M4 MacMini, and an

[5] https://www.henle.de/Levels-of-Difficulty/
[6] https://github.com/meinardmueller/synctoolbox

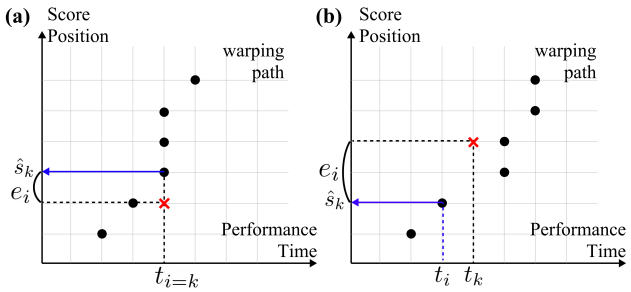

Figure 3. Two examples of error calculation using the mapping function. (a) shows a one-to-many alignment at the evaluation point, while (b) illustrates a skipped alignment.

Apple M2 Pro MacBook, with the reported latency values averaged across these devices.

4.3 Preprocessing

In the preprocessing step (see Fig. 1), the symbolic scores are synthesized to audio using FluidSynth, provided by *partitura*. Since MusicXML often lacks tempo markings, we set the synthesis tempo to each performance's average—rounded to the nearest 20 BPM—assuming performers follow approximate tempo indications.

To generate beat annotations for the synthesized score audio, we computed beat positions using the synthesis tempo and the score's time signature. For compound meters (e.g., 6/8, 9/8, and 12/8), we adopted (n)ASAP's beat annotation rules—counting them as two, three, and four beats per measure, respectively—across all datasets to align score-side annotations with performance annotations. Based on the synthesized audio, we then extract the feature using the same `Processor` used in the online phase, but precompute them offline for the entire score sequence.

5. EVALUATION

Evaluating score following is challenging due to causality, timing precision, and output latency. Since the MIREX challenge [23] provided foundational metrics, later studies introduced alternative evaluation strategies including beat-level evaluations or asynchrony [3], reflecting the task's frequent integration with automatic accompaniment systems.

In this work, we adopt two complementary evaluation perspectives. First, we evaluate in the performance domain, where errors are measured in milliseconds based on ground-truth annotations aligned to the audio. This approach is commonly used in audio-to-score alignment research and enables precise, frame-level evaluation, since the annotations directly reflect the actual timing of the performance. Second, we also evaluate in the score domain measured in beat units as suggested in [29, 57], which better reflects the nature of score following as a task of predicting the corresponding score position at each moment of the performance.

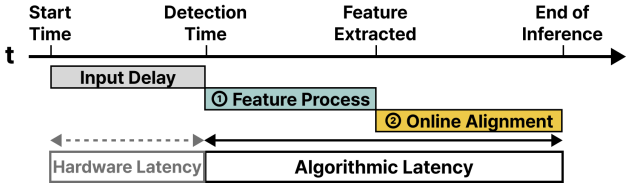

Figure 4. Defined delay types of the system. Only system delay is considered in the experiment.

5.1 Evaluation Metrics

We select evaluation metrics mostly adapted from score following MIREX benchmark [23] and audio-to-score alignment (ASA) metrics [57]. We use **Alignment Rate** (AR) within a tolerance range of $|\theta_e|$, varying from 50 ms to 2000 ms. We also compute **Absolute Errors** (AE), both in milliseconds and in beats, from which we derive the **Average Absolute Error** (AAE) and **Median Absolute Error** (MAE), along with the standard deviation σ_e. To further characterize the distribution of errors, we report **kurtosis** and **skewness** which capture the peakedness and asymmetry of the non-absolute error distribution, respectively. In addition, we report the **average latency** μ_{lat}, defined as the *system delay* from the detection time to the end of inference. Unlike total latency, this excludes *hardware latency* and is composed of two parts: (i) feature processing and (ii) execution of the online alignment algorithm for each frame step (see Fig. 4). Errors exceeding 2 seconds (or 2 beats in the score domain) are excluded from AE calculations, including both AAE and MAE, to avoid distortion from unbounded tracking failures. We report AR in two ways. The averaged piece-wise AR is a common measure, while the total AR reflects the proportion of successfully aligned beat events across the entire dataset. The latter avoids overrepresentation of shorter pieces and provides a more balanced view of overall performance.

To evaluate runtime latency under simulation, we measure two components: the average duration for extracting features from incoming audio frames, and the time taken by the alignment process to consume features and predict score positions. Specifically, the latency was computed from the moment audio was read to the time the score position was predicted—excluding hardware I/O delays. This two-step measurement allows for standardized latency reporting independent of the hardware setup.

5.2 Alignment Mapping Function

Given the alignment path, the alignment mapping function is applied to transfer the beat positions on one axis (either performance or score) to another axis to compute the alignment error. Due to the local, stepwise nature of real-time alignment, the resulting path is not necessarily monotonic and may contain multiple correspondents or skipped positions, depending on the implementation and purpose of the methods. Unlike linear interpolation methods commonly used in offline audio-to-score alignment, which assume continuous mappings, our evaluation relies only on

Dataset	Method	AAE(ms)↓ ± σ	MAE(ms)↓	Skew.	Kurt.	Piece-wise AR (%) ↑					Total AR↑ (≤2000ms, %)
						≤50ms	≤100ms	≤500ms	≤1000ms	≤2000ms	
(n)ASAP	OLTWDixon	189.55 ± 281.55	97.09	3.20	17.97	40.3	58.5	82.5	88.3	92.0	89.4
	OLTWArzt	183.56 ± 263.95	**91.18**	0.75	11.79	44.1	58.3	84.8	92.0	95.1	**92.8**
	HMM	487.73 ± 423.27	346.01	0.18	3.33	15.6	22.2	37.5	43.8	43.8	43.8
Batik	OLTWDixon	186.97 ± 262.55	104.40	3.75	24.70	28.2	51.7	82.1	85.2	87.6	89.4
	OLTWArzt	193.36 ± 269.13	107.15	1.00	12.63	35.9	53.0	82.2	87.4	90.3	89.7
	HMM	693.63 ± 376.58	641.77	0.11	0.98	4.5	10.8	34.0	46.2	64.2	61.9
Vienna4x22	OLTWDixon	285.43 ± 390.82	**132.73**	1.57	5.90	26.6	43.2	72.4	80.0	85.5	82.5
	OLTWArzt	300.41 ± 368.70	152.51	0.50	3.93	33.2	44.5	73.3	84.3	86.7	**86.7**
	HMM	439.64 ± 427.02	319.13	0.15	3.79	23.5	33.3	51.1	57.1	63.0	75.9

Table 2. Evaluation results on three datasets using different score-following methods. The piece-wise alignment rate (AR) is measured as the average over pieces, while the total AR indicates the global proportion of aligned beat events across the entire dataset. All tests were conducted with STFT-based Chroma as features.

predictions made prior to or at each evaluation time point. To reflect this, we define the mapping function as follows:

$$\hat{u}_k = \min\{\, u_i \mid (u_i, v_i) \in \mathcal{W},\ v_i = \max\{v_j \mid v_j \le k\}\},$$

where $\mathcal{W} = \{(u_i, v_i)\}$ is the warping path expressed in the frame indices: u_i is the score-rendered-audio frame index and v_i is the performance-audio frame index. The inner $\max$ finds the latest performance frame v_i not exceeding the current frame k, and the outer $\min$ selects the smallest score frame u_i among those alignments. This mapping relies solely on past or current frames to maintain causality. It handles skipped or one-to-many mappings and avoids any interpolation methods that depend on future frames.

6. RESULTS

Table 2 presents a comparison of alignment methods based on performance-domain evaluation, measured in milliseconds. All methods exhibit positive skewness in error distribution, reflecting the expected lag of the beat estimates in real-time alignment. The overall results show that the OLTW-based method outperforms the HMM baseline across all datasets in both alignment accuracy and coverage. While OLTWDixon and OLTWArzt show comparable MAE depending on the dataset, OLTWArzt consistently achieves higher coverage (*Total AR*), suggesting that it is more robust against overall failures. The difference likely stems from OLTWDixon skipping uncertain regions, while OLTWArzt's "backward-forward" strategy corrects early misalignments and enhances coverage. Despite having the lowest AR, the HMM shows the lowest skewness and kurtosis primarily because significant errors (>2 s) are excluded from the summary statistics and its "sticky" behavior to linger in the same state in local regions tends to narrow the error distribution.

Table 3 presents an evaluation comparison in beat units, offering a tempo-normalized perspective. The overall trend mirrors the performance-domain results in millisecond, but these results are standardized across tempi. AAE remains around 0.3 beats, with median values typically below 0.2. Total AR is consistently lower than the 2000 ms-based metric, reflecting that most pieces have tempi above 60BPM, where two beats span less than two seconds.

Dataset	Method	AAE↓(beats) ± σ	MAE↓(beats)	AR↑ (%)
(n)ASAP	OLTWDixon	0.22 ± 0.27	0.13	83.4
	OLTWArzt	0.27 ± 0.30	0.16	85.2
	HMM	0.80 ± 0.54	0.66	76.9
Batik	OLTWDixon	0.20 ± 0.27	0.11	88.9
	OLTWArzt	0.29 ± 0.34	0.18	88.8
	HMM	0.80 ± 0.38	0.67	59.3
Vienna4x22	OLTWDixon	0.31 ± 0.33	0.19	78.3
	OLTWArzt	0.37 ± 0.38	0.24	84.0
	HMM	0.76 ± 0.78	0.51	70.3

Table 3. Beat-level evaluation results including total alignment rate (AR) (%).

Feature Process			Online Alignment	
Type	MAE (ms)	Latency (ms)	Method	Latency (ms)
Chroma	265.50	3.05	OLTWDixon	1.22
mel	297.92	3.40	OLTWArzt	0.07
CQT	341.25	42.58	HMM	3.59
LSE	241.85	0.91		
MFCC	931.81	2.58		

Table 4. Comparison of feature types and alignment methods in terms of alignment error (MAE) and latency. LSE is log-spectral energy feature that was adopted in [38]. Latency values are averaged over the hardware setups evaluated in Section 4.

In addition, a comparison of various feature types and latencies of the alignment methods are reported in Table 4. Among the features, log-spectral energy (LSE) shows the lowest MAE (241.85 ms) and delay (0.91 ms), indicating strong performance with minimal overhead. In contrast, CQT and MFCC yield higher MAE, with CQT also requiring considerable extraction time (42.58 ms), which limits its real-time suitability. For alignment methods, OLTWArzt achieves the lowest latency (0.07 ms), whereas HMM shows noticeably higher delay (3.59 ms) due to its computational complexity. These results highlight a trade-

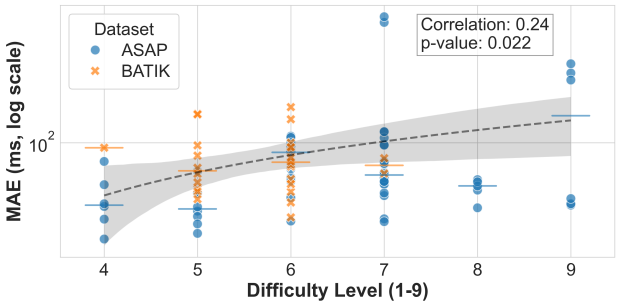

Figure 5. A scatter plot of mean absolute error (MAE) and Henle's difficulty level in (n)ASAP and Batik dataset. The MAE results are from OLTWArzt.

off between alignment accuracy and runtime efficiency, with LSE and OLTWArzt providing a favorable balance for low-latency use.

The results also show about the characteristics of the datasets. While the overall alignment performance between (n)ASAP and Batik is comparable, Vienna4x22 shows noticeably higher error variance and kurtosis. This reflects the dataset's unique structure—22 diverse renditions for each of only four pieces—leading to substantial variability in expressive timing, articulation, and interpretation. These variations present additional challenges for score following and result in heavier-tailed error distributions, as seen in the higher kurtosis values.

7. DISCUSSIONS

Figure 5 further illustrates the relationship between musical difficulty and alignment accuracy for (n)ASAP and Batik. We observe a moderate positive correlation ($r = 0.24$, $p = 0.022$) between MAE and the annotated difficulty levels, indicating that technically more demanding pieces tend to produce larger alignment errors. Vienna4x22 was excluded from this analysis due to its use of short excerpts, which makes consistent difficulty grading unreliable.

To further understand how alignment behaviors differ from methods, Figure 6 illustrates an example of alignment result comparing OLTWArzt (left) and HMM (right). Although OLTWArzt smoothly follows the beat events, the HMM warping path shows frequent horizontal segments, indicating the "sticky" tendency to stay near note onsets, reflecting its state-based formulation that emphasizes onset transitions. This leads to cases where it lingers on sustained notes and becomes locally stuck, showing limited forward momentum. The corresponding region (highlighted in yellow) exhibits changes in harmony, note density, and dynamics compared to the preceding passage, which provides sufficient contrast for the score follower to recover.

Lastly, we found that not only the choice of evaluation metrics, but also how alignment errors are computed (Section 5.2) can affect accuracy results to a meaningful extent. Small differences in error calculation sometimes led to noticeable shifts in reported accuracy.

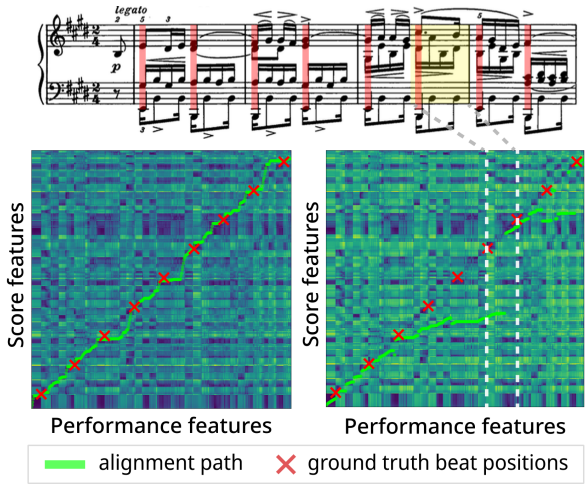

Figure 6. Two examples of alignment path with beat positions: (left) OLTWArzt, (right) HMM.

8. USE CASES AND APPLICATIONS

To demonstrate the practicality of our package, we built a lightweight web application that runs locally with real-time audio input or pre-recorded files. Built with websocket-based communication, the system responds quickly enough to ensure minimal perceptual delay. Our companion website includes a video demonstration and a link to the source code. This application aims to help researchers test their own score following models in an interactive setting. Beyond the web demo, our package is also used as the score following module in the ACCompanion [3], a real-time accompaniment system. These applications demonstrate the versatility of our framework and validate its utility in interactive music scenarios.

9. CONCLUSIONS AND FUTURE WORK

We presented a systematic framework for real-time audio-based score following as the open-source Python package *Matchmaker*. It supports live and simulation-based evaluation with baseline models, enabling reproducible benchmarking across datasets and features. Experiments on three public piano datasets show that the OLTWArzt variant achieves the highest performance and that the onset-sensitive spectral feature (LSE) outperforms chroma in both accuracy and latency. However, the current framework is limited in its support for tempo models commonly integrated with HMM-based score followers which may partly explain the limited performance of the HMM baseline. Also, recent works often include learned features or multimodal input which poses a new challenge to evaluate. Although our evaluation was limited to classical piano, extending *Matchmaker* to other instruments and genres requires only adapting the proper datasets and feature extraction modules. Future work will extend the framework to support a wider variety of instruments and musical styles, and include additional feature representations, advanced tempo modeling, and multimodal inputs.

10. ACKNOWLEDGMENTS

This work has been supported by the Austrian Science Fund (FWF), grant agreement PAT 8820923 ("*Rach3: A Computational Approach to Study Piano Rehearsals*"). Additionally, this work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) under Grant RS-2023-NR077289.

11. REFERENCES

[1] R. B. Dannenberg, "An On-Line Algorithm for Real-Time Accompaniment," in *Proceedings of the International Computer Music Conference (ICMC '84)*, Paris, France, 1984.

[2] B. Vercoe, "The Synthetic Performer in the Context of Live Performance," in *Proceedings of the International Computer Music Conference (ICMC '84)*, Paris, France, 1984.

[3] C. Cancino-Chacon, S. Peter, P. Hu, E. Karystinaios, F. Henkel, F. Foscarin, N. Varga, and G. Widmer, "The ACCompanion: Combining Reactivity, Robustness, and Musical Expressivity in an Automatic Piano Accompanist," in *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI-23)*, Macao, SAR, China, May 2023, arXiv:2304.12939 [cs, eess]. [Online]. Available: http://arxiv.org/abs/2304.12939

[4] K. Armstrong, T.-C. Hung, J.-X. Huang, and Y.-W. Liu, "Real-time piano accompaniment model trained on and evaluated according to human ensemble characteristics," in *Proceedings of the Sound and Music Computing (SMC)*, Porto, Portugal, 2024.

[5] C. Raphael, "Music Plus One and Machine Learning," in *Proceedings of the 27th International Conference on Machine Learning (ICML 2010)*, Haifa, Israel, 2010.

[6] A. Maezawa, "I Got Rhythm, so Follow Me More: Modeling Score-Dependent Timing Synchronization in a Piano Duet," in *Proceedings of the Sound and Music Computing Conference (SMC 2024)*, Porto, Portugal, 2024.

[7] A. Arzt, G. Widmer, and S. Dixon, "Automatic Page Turning for Musicians via Real-Time Machine Listening," in *Proceedings of the European Conference on Artificial Intelligence (ECAI)*, 2008.

[8] F. Henkel, S. Schwaiger, and G. Widmer, "Fully Automatic Page Turning on Real Scores," in *Extended Abstracs for the Late-Breaking Demo Session of the 22nd International Society for Music Information Retrieval Conference (ISMIR 2021)*, Online, 2021, arXiv:2111.06643 [cs]. [Online]. Available: http://arxiv.org/abs/2111.06643

[9] C. Brazier and G. Widmer, "Towards Reliable Real-time Opera Tracking: Combining Alignment with Audio Event Detectors to Increase Robustness," in *Proceedings of the Sound and Music Computing Conference*, Online, 2020, arXiv:2006.11033 [cs, eess]. [Online]. Available: http://arxiv.org/abs/2006.11033

[10] J. Park, S. Yong, T. Kwon, and J. Nam, "A Real-Time Lyrics Alignment System Using Chroma and Phonetic Features for Classical Vocal Performance," in *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Seoul, Korea, Republic of: IEEE, Apr. 2024, pp. 1371–1375. [Online]. Available: https://ieeexplore.ieee.org/document/10445926/

[11] R. Gong, P. Cuvillier, N. Obin, and A. Cont, "Real-time audio-to-score alignment of singing voice based on melody and lyric information," in *Interspeech 2015*. ISCA, Sep. 2015, pp. 3312–3316. [Online]. Available: https://www.isca-archive.org/interspeech_2015/gong15_interspeech.html

[12] T. Otsuka, K. Nakadai, T. Takahashi, T. Ogata, and H. G. Okuno, "Real-Time Audio-to-Score Alignment Using Particle Filter for Coplayer Music Robots," *EURASIP Journal on Advances in Signal Processing*, vol. 2011, no. 1, p. 384651, Dec. 2011. [Online]. Available: https://asp-eurasipjournals.springeropen.com/articles/10.1155/2011/384651

[13] O. Lartillot, C. Cancino-Chacon, and C. Brazier, "Real-Time Visualisation of Fugue Played by a String Quartet," in *Proceedings of the Sound and Music Computing Conference (SMC 2020)*, Online, 2020.

[14] R. Dannenberg and B. Mont-Reynaud, "Following an Improvisation in Real Time," in *Proceedings of the International Computer Music Conference*, Champaign/Urbana, Illinois, USA, 1987.

[15] R. B. Dannenberg and H. Mukaino, "New Techniques for Enhanced Quality of Computer Accompaniment," in *Proceedings of the International Computer Music Conference*, Cologne, Germany, 1988.

[16] B. Vercoe and M. Puckette, "Synthetic Rehearsal: Training the Synthetic Performer," in *Proceedings of the International Computer Music Conference (ICMC '85)*, Vancouver, BC, Canada, 1985.

[17] M. Puckette, "Score following using the sung voice," in *Proceedings of the International Computer Music Conference (ICMC '95)*, Banff, AB, Canada, 1995.

[18] P. Desain, H. Honing, and H. Heijink, "Robust Score-Performance Matching: Taking Advantage of Structural Information," in *Proceedings of the International Computer Music Conference (ICMC)*, Thessalonki, Greece, 1997.

[19] C. Raphael and Y. Gu, "Orchestral Accompaniment for a Reproducing Piano," in *Proceedings of the International Computer Music Conference (ICMC 2009)*, Montreal, Canada, 2009.

[20] M. Prockup, D. Grunberg, A. Hrybyk, and Y. E. Kim, "Orchestral Performance Companion: Using Real-Time Audio to Score Alignment," *IEEE MultiMedia*, vol. 20, no. 2, pp. 52–60, Apr. 2013. [Online]. Available: http://ieeexplore.ieee.org/document/6530591/

[21] A. Arzt and G. Widmer, "Real-Time Music Tracking Using Multiple Performances as a Reference," in *Proceedings of the 16th International Society for Music Information Retrieval Conference (ISMIR 2015)*, Malaga, Spain, 2015.

[22] F. Henkel, S. Balke, M. Dorfer, and G. Widmer, "Score Following as a Multi-Modal Reinforcement Learning Problem," *Transactions of the International Society for Music Information Retrieval*, vol. 2, no. 1, pp. 67–81, Nov. 2019. [Online]. Available: http://transactions.ismir.net/articles/10.5334/tismir.31/

[23] A. Cont, D. Schwarz, N. Schnell, and C. Raphael, "Evaluation of Real-Time Audio-to-Score Alignment," in *Music Information Retrieval Evaluation eXchange (MIREX 2007)*, Vienna, Austria, 2007.

[24] A. Cont, "Antescofo: Anticipatory Synchronization and Control of Interactive Parameters in Computer Music," in *Proceedings of the International Computer Music Conference (ICMC '08)*, Belfast, Ireland, 2008.

[25] J. Echeveste, P. Cuvillier, and A. Cont, "Improved Synchronization of a Pre-Recorded Music Accompaniment on a User's Music Playing," U.S. Patent US 2023/0 082 086 A1, Mar., 2023.

[26] S. Dixon and G. Widmer, "MATCH: A Music Alignment Tool Chest," in *Proceedings of the 6th International Conference on Music Information Retrieval (ISMIR 2005)*, London, UK, 2005.

[27] C. Joder, S. Essid, and G. Richard, "Learning Optimal Features for Polyphonic Audio-to-Score Alignment," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 10, pp. 2118–2128, Oct. 2013. [Online]. Available: http://ieeexplore.ieee.org/document/6525340/

[28] M. Muller, F. Kurth, and T. Roder, "Towards an Efficient Algorithm for Automatic Score-to-Audio Synchronization," in *Proceedings of the 5th International Conference on Music Information Retrieval (ISMIR 2004)*, Barcelona, Spain, 2004.

[29] Z. Duan and B. Pardo, "A state space model for online polyphonic audio-score alignment," in *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Prague, Czech Republic: IEEE, May 2011, pp. 197–200. [Online]. Available: http://ieeexplore.ieee.org/document/5946374/

[30] P.-W. Chou, F.-N. Lin, K.-N. Chang, and H.-Y. Chen, "A Simple Score Following System for Music Ensembles Using Chroma and Dynamic Time Warping," in *Proceedings of the 2018 ACM on International Conference on Multimedia Retrieval*. Yokohama Japan: ACM, Jun. 2018, pp. 529–532. [Online]. Available: https://dl.acm.org/doi/10.1145/3206025.3206090

[31] M. Muller, "Music Synchronization," in *Fundamentals of Music Processing*. Cham: Springer International Publishing, 2021, pp. 119–170. [Online]. Available: https://link.springer.com/10.1007/978-3-030-69808-9_3

[32] M. Pérez Fernández, H. Kirchhoff, and X. Serra, "A comparison of pitch chroma extraction algorithms," in *Proceedings of the 19th Sound and Music Computing Conference (SMC/JIM/IFC 2022)*. Saint-Étienne, France: SMC Network, 2022. [Online]. Available: https://doi.org/10.5281/zenodo.6573082

[33] C.-T. Chen, J.-S. R. Jang, W.-S. Liu, and C.-Y. Weng, "An efficient method for polyphonic audio-to-score alignment using onset detection and constant Q transform," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Shanghai: IEEE, Mar. 2016, pp. 2802–2806. [Online]. Available: http://ieeexplore.ieee.org/document/7472188/

[34] J. J. Carabias-Orti, F. J. Rodriguez-Serrano, P. Vera-Candeas, and N. Ruiz-Reyes, "An Audio to Score Alignment Framework Using Spectral Factorization and Dynamic Time Warping," in *Proceedings of the 16th International Society for Music Information Retrieval Conference (ISMIR 2015)*, Malaga, Spain, 2015.

[35] F. Korzeniowski and G. Widmer, "Refined Spectral Template Models for Score Following," in *Proceedings of the Sound and Music Computing Conference (SMC 2013)*, Stockholm, Sweden, 2013.

[36] S. Ewert, M. Muller, and P. Grosche, "High resolution audio synchronization using chroma onset features," in *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2009, pp. 1869–1872.

[37] A. Pillay, "A Neural Score Follower for Computer Accompaniment of Polyphonic Musical Instruments," Master's thesis, Carnegie Mellon University, Pittsburgh, PA, USA, 2024.

[38] S. Dixon, "An On-Line Time Warping Algorithm for Tracking Musical Performances," in *Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI 05)*, Edinburgh, Scotland, 2005.

[39] R. Macrae and S. Dixon, "Polyphonic Score Following Using on-Line Time Warping," in *Music Information Retrieval Evaluation eXchange (MIREX 2008)*, Philadelphia, USA, 2008.

[40] F. J. Rodriguez-Serrano, P. Vera-Candeas, and J. J. Carabias-Orti, "A Real-Time Score Follower for MIREX 2015," in *Music Information Retrieval Evaluation eXchange (MIREX 2015)*, Malaga, Spain, 2015.

[41] J. J. Carabias, F. J. Rodriguez, and P. Vera, "A Real-Time Nmf-Based Score Follower for MIREX 2012," in *Music Information Retrieval Evaluation eXchange (MIREX 2012)*, Porto, Protugal, 2012.

[42] A. Arzt, "Real-Time Music Tracking Using Tempo-Aware on-Line Dynamic Time Warping," in *Proceedings of the Vienna Talk on Musical Acoustics (VITA)*, Vienna, Austria, 2010.

[43] P. Cano, A. Loscos, and J. Bonada, "Score-Performance Matching using HMMs," in *Proceedings of the International Computer Music Conference*, Beijing, China, 1999.

[44] E. Nakamura, P. Cuvillier, and A. Cont, "Autoregressive Hidden Semi-Markov Model of Symbolic Music Performance for Score Following," in *Proceedings of the 16th International Society for Music Information Retrieval Conference (ISMIR 2015)*, Malaga, Spain, 2015. [Online]. Available: https://archives.ismir.net/ismir2015/paper/000015.pdf

[45] C. Raphael, "A Bayesian Network for Real-Time Musical Accompaniment," in *Proceedings of the 14th International Conference on Neural Information Processing Systems*, Vancouver, BC, Canada, 2001, pp. 1433–1439. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2001/file/2b0f658cbffd284984fb11d90254081f-Paper.pdf

[46] R. Yamamoto, S. Sako, and T. Kitarmura, "Real-Time Audio to Score Alignment Using Segmental Conditional Random Fields and Linear Dynamical System," in *Music Information Retrieval Evaluation eXchange (MIREX 2012)*, Porto, Portugal, 2012.

[47] Y. Jiang and C. Raphael, "Score Following with Hidden Tempo Using a Switching State-Space Model," in *Proceedings of the 21st International Society for Music Information Retrieval Conference (ISMIR 2020)*, Online, 2020.

[48] M. Dorfer, F. Henkel, and G. Widmer, "Learning to Listen, Read, and Follow: Score Following as a Reinforcement Learning Game," in *Proceedings of the 19th International Society for Music Information Retrieval Conference (ISMIR 2018)*, Paris, France, 2018.

[49] S. Dixon, "Live Tracking of Musical Performances using On-Line Time Warping," in *Proceedings of the 8th International Conference on Digital Audio Effects (DAFx'05)*, Madrid, Spain, 2005.

[50] A. Arzt and G. Widmer, "Towards Effective Any-Time Music Tracking," in *Proceedings of the Starting AI Researchers Symposium (STAIRS), held at ECAI 2010*, Lisbon, Portugal, 2010.

[51] S. Behnel, R. Bradshaw, C. Citro, L. Dalcin, D. S. Seljebotn, and K. Smith, "Cython: The best of both worlds," *Computing in Science & Engineering*, vol. 13, no. 2, pp. 31–39, 2011.

[52] S. D. Peter, C. E. Cancino-Chacón, F. Foscarin, A. P. McLeod, F. Henkel, E. Karystinaios, and G. Widmer, "Automatic Note-Level Score-to-Performance Alignments in the ASAP Dataset," *Transactions of the International Society for Music Information Retrieval*, vol. 6, no. 1, pp. 27–42, Jun. 2023. [Online]. Available: http://transactions.ismir.net/articles/10.5334/tismir.149/

[53] P. Hu and G. Widmer, "The Batik-plays-Mozart Corpus: Linking Performance to Score to Musicological Annotations," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2023.

[54] W. Goebl, "The vienna 4x22 piano corpus," 1999. [Online]. Available: http://dx.doi.org/10.21939/4X22

[55] F. Foscarin, E. Karystinaios, S. D. Peter, C. Cancino-Chacón, M. Grachten, and G. Widmer, "The match file format: Encoding alignments between scores and performances," in *Proceedings of the Music Encoding Conference (MEC 2022)*, Halifax, Canada.

[56] C. Brazier and G. Widmer, "Addressing the Recitative Problem in Real-time Opera Tracking," in *Proceedings of Frontiers of Research in Speech and Music FRSM 2020*, Online, Oct. 2020, arXiv:2010.11013 [eess].

[57] A. Morsi and X. Serra, "Bottlenecks and solutions for audio to score alignment research." in *Proceedings of the 23rd International Society for Music Information Retrieval Conference (ISMIR 2022)*, 2022, pp. 272–279.