Magic Quill: A Real-time Score Engraving System for Live MIDI Performances

Lynn Ye
School of Music
Carnegie Mellon University

Roger Dannenberg
Computer Science Department
Carnegie Mellon University

Chris Donahue
Computer Science Department
Carnegie Mellon University

Abstract

We present **Magic Quill**, a browser-based system that visualizes live MIDI performances as an evolving engraved score. An online beat tracker produces provisional notation immediately, while an asynchronous offline model revises earlier measures as additional musical context becomes available. On consumer hardware, the online model requires approximately 0.13 ms per note and browser-based online beat-tracking inference completes within approximately 10 ms; offline refinement arrives after several seconds and achieves competitive transcription accuracy using substantially less context than prior offline systems. A demonstration of our proposed system is available at `https://magicquill.github.io`.

1 Introduction

Music unfolds as a stream of sound, while listeners often develop an understanding of its connections and structure over longer spans of time. Mapping a live performance to notation can invite audiences to "engage in a decoding process to develop a deeper understanding of the musical process" [1]. Score-following systems provide such a mapping when a reference score is available [2, 3]. They cannot, however, visualize improvisatory or otherwise scoreless performances.

An alternative approach is to construct the score as the performance unfolds. Performance MIDI-to-Score (PM2S) transcription addresses this challenge by mapping a series of MIDI events to discrete musical notation. Early PM2S methods used probabilistic models for rhythm, tempo, staff, and note-value estimation [4–6]. More recent neural systems use recurrent networks and Transformer encoder-decoders to predict comprehensive score representations [7–10]. These systems achieve strong transcription accuracy by utilizing future context, but their structural and computational latency prevents an immediate live visualization. In contrast, commercial notation tools such as Logic Pro [11] provide immediate visualization through fixed-tempo quantization, requiring performers to conform to a rigid beat grid.

Real-time PM2S also relates to interactive symbolic music systems that perform low-latency online inference from MIDI. For example, Piano Genie maps button presses to pitches using an LSTM [12]; ReaLchords generates real-time chord accompaniment [13]; and JamBot supports interactive improvisation with an anticipatory music model [14]. These systems generate or transform musical material, whereas our aim is to construct an evolving notational interpretation of a performance.

To this end, we present Magic Quill, a browser-based interface that visualizes live MIDI piano performances as an evolving score. A lightweight online model creates provisional notation immediately, while an asynchronous offline model revises earlier measures as more musical context becomes

Submitted to 40th Conference on Neural Information Processing Systems (NeurIPS 2026). Do not distribute.

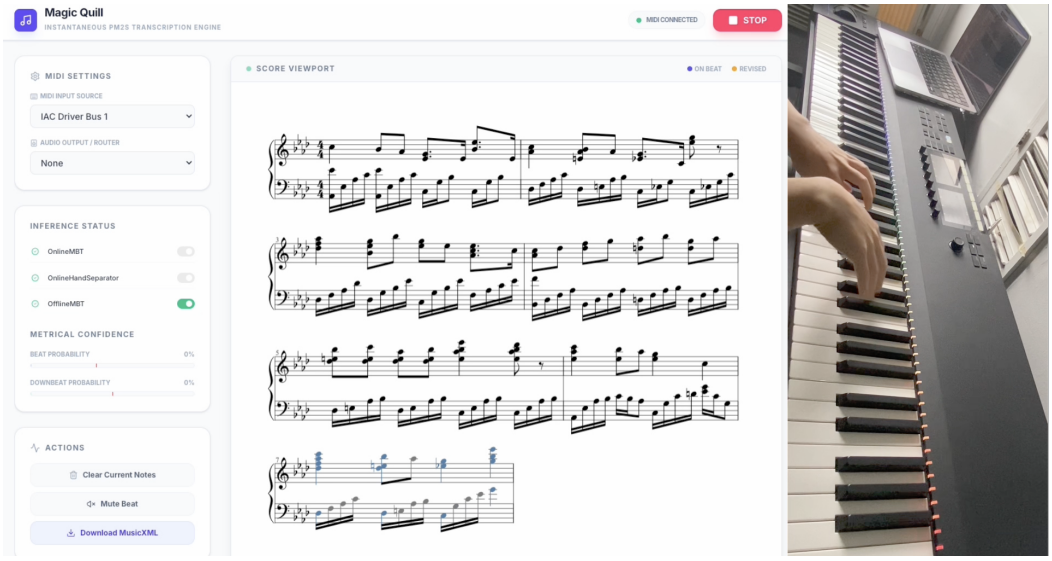


Figure 1: Magic Quill during a live MIDI performance. Provisional notes appear immediately in gray and blue; asynchronous offline refinement revises history and renders stabilized notes in black.

available. This dual-timescale design delivers an immediate visual response as well as a more accurate transcription after refinement. Figure 1 shows a screenshot of the system as a performer plays a MIDI keyboard. The accompanying video demonstration is available at the link in the abstract.

2 Magic Quill System

Magic Quill is a web-based application for real-time score engraving. Once a user connects a MIDI keyboard and starts recording, the system predicts a metrical position for each incoming note event and renders the result in the score viewport. During the performance, the interface also separately displays the model's beat and downbeat confidence within approximately 10 ms.

The user first sees provisional measures containing gray and blue notes produced by the online model. Their color signals that the online model operates without future context and that its output may therefore be revised. To limit frequent visual rearrangement, the interface updates the score only when the online model detects a beat. After several seconds, once sufficient note context has accumulated, the offline model revises the provisional measures. Refined notes are rendered in black to represent a more context-informed score transcription of the performance. The result is a dual-timescale interface: performed notes appear immediately, while the recent musical history is periodically replaced by a more accurate score transcription result.

The system works by first quantizing the onset of each note-on event. An online model derives provisional beat labels for immediate engraving, and an offline model later refines the metrical position of the same events. A shared post-processing stage then derives the remaining score attributes, including note duration, staff assignment, meter, and key. The resulting note representation is converted to MusicXML and rendered in the score viewport using OpenSheetMusicDisplay [15].

2.1 Two-stage Onset Quantization

We predict the metrical position of each note-on event through a two-stage process that runs online inference and offline refinement asynchronously.

2.1.1 Online Model

Through the MIDI protocol, a performance is captured as a stream of N note events. We represent this stream as $X = \{x_1, x_2, \ldots, x_N\}$, where each note $x_i = (t_i, p_i)$ is a pair containing its onset time and pitch. The system ultimately infers a corresponding sequence of metrical positions, one for

each note. For online processing, a neural model first predicts whether each incoming note coincides with a beat or downbeat. Detected beats establish a provisional tempo grid, and linear interpolation maps the remaining note onsets to exact positions on that grid. The model receives the proxy features $\hat{x}_i = (p_i, \log(1 + t_i - t_{i-1}))$, where $t_i - t_{i-1}$ is the inter-onset interval (IOI) from the preceding note. For low latency and simplicity, we use a unidirectional LSTM, which has been demonstrated as a practical choice for note-level real-time prediction problems [12].

2.1.2 Offline Model

The offline tracker provides a high-resolution refinement of the online results. Given an input sequence X, it uses bidirectional performance context to infer metrical positions $Y' = \{y'_i\}_{i=1}^{N}$. Each position $y'_i = (a_i, u_i)$ contains a tactus token $a_i \in \{0, \ldots, 4\}$, representing the beat index within a measure, and a tatum token $u_i \in \{0, \ldots, 23\}$, representing the subdivision within that beat. We model $P(Y'|X) = \prod_{i=1}^{N} P(y'_i \mid X, y'_{<i})$ with an encoder-decoder Transformer [16]. Its input is an interleaved sequence of pitch and inter-onset interval (IOI): pitch is tokenized into 88 classes, and log-transformed IOI into 257 bins. The model uses six encoder layers, three decoder layers, a model dimension of 256, pre-normalization [17], and rotary positional encoding in every attention layer [18], for a total of 8.1M parameters.

Predicting one metrical position for every note-on event establishes a persistent one-to-one mapping between the performed pitches and the generated score. This mapping preserves every performed pitch and enables the offline model to revise a specific region without losing correspondence with the provisional output. In contrast, models trained with coarser alignments may produce ambiguous boundaries or a different number of score events, complicating retrospective replacement.

2.2 Post-processing Method

We employ a suite of heuristics to derive the rest of the final engraved score attributes from metrical predictions. To calculate **note value** (the written duration, e.g., quarter note vs. eighth note), we map each MIDI note-off event to the closest quantized onset position on the metrical grid. We reconstruct the **time signature** by first grouping the quantized positions (a_i, u_i) into measures, then determine the numerator by the maximum tactus index within a measure. To determine **staff assignment** (treble vs. bass clef), we implement a dynamic online clustering algorithm, where each incoming pitch p_i is assigned to one of the two cluster centroids c_i that minimizes the distance $|p_i - c_i|$. Following assignment, the corresponding centroid is updated via exponential moving average, allowing the centers to shift dynamically as the performance spans different registers of the keyboard. For **key signature** estimation, we use the Krumhansl-Schmuckler algorithm [19] on the entire performance.

2.3 Web Development

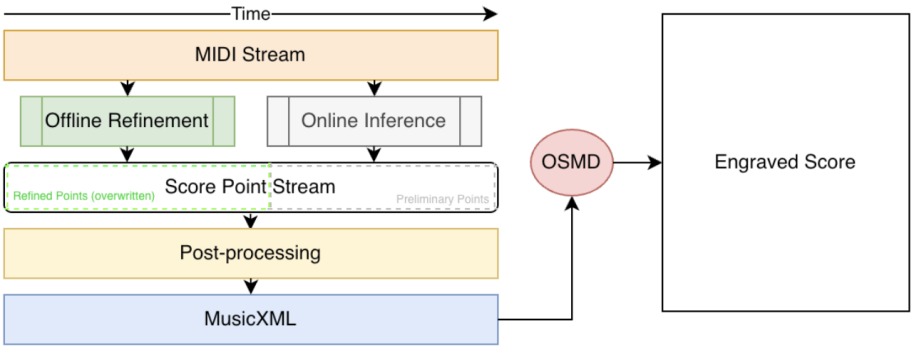


Figure 2: Magic Quill system architecture. Online and offline predictions update a shared score-point stream, which is post-processed, converted to MusicXML, and rendered in the browser with OpenSheetMusicDisplay (OSMD).

To integrate online and offline predictions in the web application, we define a **score-point stream**: a list of objects holding the pitch, metrical onset, duration, and staff assignment of each performed

MIDI event (Figure 2). Because note-level alignment is preserved, the system can identify the exact region processed by offline refinement and overwrite only the corresponding provisional metrical positions.

The score-point stream is passed to a unified post-processing algorithm for meter, measure boundary, and key signature prediction. Any revision is therefore propagated through the same engraving pipeline. The processed note list is serialized, converted to MusicXML, and sent to OpenSheetMusicDisplay (OSMD) [15] for browser-based engraving.

The front end is implemented in React, Web MIDI handles input routing, and ONNX Runtime provides cross-platform browser inference. We use key-value caching [20] to accelerate the offline Transformer. This local implementation avoids a network dependency and allows the system to operate on consumer hardware.

3 Experiments

We train our models using the ACPAS [21] and (n)ASAP [22] datasets. Both consist primarily of classical piano performances and therefore introduce a substantial stylistic bias. The accompanying demonstration, however, includes preliminary examples of pop and swing performances outside this training distribution, showing initial promise of the model generalizing to out-of-domain pieces.

The online and offline models minimize binary cross-entropy and cross-entropy, respectively, using AdamW [23] with learning rates of $3\mathrm{e}{-4}$ and $1\mathrm{e}{-4}$, linear warmup, and cosine decay. We sample segments of up to 512 notes for the LSTM and 256 notes for the Transformer. Augmentation includes pitch shifts of ± 12 semitones, tempo scaling in $[0.8, 1.2]$, and note dropping with probability 0.15; Transformer training also uses IOI-dependent Gaussian onset jitter. The models are trained for 54,000 and 33,000 steps, respectively. Online evaluation follows the ACPAS split, while offline evaluation follows the ASAP split of Beyer et al. [9].

We evaluate the two system timescales separately. First, we compare the online beat tracker with an offline LSTM-based tracker. We then use conventional PM2S metrics to evaluate offline refinement and the complete hybrid system.

3.1 Online Beat Tracking

We evaluate our online beat tracker using the ACPAS dataset [21], adopting the standard ± 70 ms tolerance window for true positives [24]. We also simulate a scenario in which a user provides occasional downbeat *hints*, such as tapping a pedal at the beginning of a measure.

Table 1 compares our model, with and without downbeat hints, with the offline beat tracker of Liu et al. [8]. Without hints, our online model attains 80.2% and 83.2% of the reference model's beat and downbeat F1 scores. Four downbeat hints increase these ratios to 80.9% and 86.9%, respectively. Given our model's online constraints and low inference latency (see §3.2 for MacBook M1 Pro benchmarks), this represents a competitive trade-off for real-time applications.

	Beat			**Downbeat**		
Model	Precision	Recall	F1	Precision	Recall	F1
Ours	**77.4**	71.0	74.1	58.3	65.6	61.7
Ours (4 db hints)	77.3	**72.4**	**74.8**	**61.4**	**68.0**	**64.5**
Liu et al. [8]	95.0	89.9	92.4	74.3	74.2	74.2

Table 1: Beat and downbeat tracking results (Precision, Recall, and F1, %). Liu et al.'s method is reported separately because it uses the full performance context for beat prediction. Bold indicates the best result among our methods.

3.2 Offline Refinement

To evaluate our hybrid PM2S system more holistically, we employ MUSTER [7, 25], a conventional metric which uses music alignment algorithms to map transcribed scores to ground-truth references, deriving edit-distance-based error rates.

To characterize operation on consumer hardware, we benchmark the real-time factor (RTF) and latency of each model on a MacBook M1 Pro. RTF is the ratio of input MIDI duration to processing time. Table 2 also reports latency in seconds. For the online model this is per-note inference time, while for offline models it includes the delay required to accumulate a chunk before inference. We report "worst-case" performance as the mean of the bottom 5% of RTF values and top 5% of latency values. The approximately 10 ms browser inference time reported in Section 2 measures online beat-tracking inference in the web application; it does not measure the full delay from MIDI input to visible notation.

	MUSTER (↓)						**RTF** (↑)		**Latency (s)** (↓)	
Method	$\mathcal{E}_{\text{p}}$	$\mathcal{E}_{\text{miss}}$	$\mathcal{E}_{\text{extra}}$	$\mathcal{E}_{\text{onset}}$	$\mathcal{E}_{\text{off.}}$	$\mathcal{E}_{\text{avg}}$	Mean	Worst	Mean	Worst
Logic Pro* [11]	**2.18**	7.21	9.38	28.41	49.31	19.30	–	–	–	–
Beyer et al. [9]	3.11	7.56	**6.44**	15.55	**23.84**	**11.30**	14.8	5.9	51.3	105
Online (proposed)	2.19	7.51	9.46	31.74	37.06	17.59	**1026.8**	**521.8**	**1.29e-4**	**1.41e-4**
Offline (proposed)	2.31	**6.46**	8.53	**15.15**	30.21	12.53	30.6	8.8	7.24	16.2
Hybrid (proposed)	2.31	**6.46**	8.53	**15.15**	30.21	12.53	**1026.8**	**521.8**	**1.29e-4**	**1.41e-4**

Table 2: Comparative quantitative evaluation on Beyer et al.'s ASAP test set. ↑/↓ indicate the direction of improvement. *Logic Pro uses Smart Tempo, which relies on offline global information to estimate tempo. Real-time factor (RTF) is calculated by dividing performance duration by inference time. Latency is reported in seconds: online latency is per-note inference time, while offline latency includes chunk accumulation. The Hybrid row reports the latency of its initial online estimate; its subsequent refinement has the latency shown in the Offline row. All Transformer-based models use key-value caching [20]. Bold values denote the best results, while underlined values denote the second-best results.

Results. Table 2 shows that the hybrid system provides an online estimate in approximately 0.13 ms per note and reaches the offline model's average MUSTER error of 12.53. The offline model obtains the lowest observed onset error in our comparison (15.15, compared with 15.55 for Beyer et al.), while requiring substantially less context and fewer parameters. Its overall error remains higher than Beyer et al. (12.53 versus 11.30), primarily because note duration and other score attributes are derived heuristically. Time-signature estimation also remains challenging: exposure bias during decoding can produce a locally plausible rhythmic grid with an incorrect meter. In many use cases, the time signature could be provided as a hint to eliminate estimation errors.

4 Demonstration and Discussion

Figure 1 shows the interface during the first performance in the accompanying demonstration. A performer improvises a passage in 4/4 meter on a MIDI keyboard while Magic Quill constructs the score. Newly performed notes appear provisionally in gray and blue; after sufficient context accumulates, the offline model revises the recent history and the interface redraws refined notes in black. The video hence presents not only a transcription result but also how the system arrives at it.

The engraved representation exposes musical relationships that may be difficult to discern from a single hearing. In this example, the separation between staves makes the left-hand texture and the overall bass line visible, while bar lines organize the rhythmic pulse into measure-level groups. Accidentals and chord spelling provide additional cues for special harmonies. These observations illustrate how score transcription can make musical relationships visible, which is one motivation for high-quality PM2S.

Agency. At inference time, Magic Quill is reactive: it neither introduces musical events nor initiates changes to a performance. The performer determines the pitches, timing, dynamics, and expressive

gestures, while the system decides how those events are represented through rhythm, meter, duration, staff, and key. Showing both provisional notation and retrospective revision makes this interpretive role visible.

Although the present demonstration is primarily audience-facing, the evolving score could also influence performer agency by acting as an external musical memory. A solo improviser might revisit an earlier motif, vary it, or deliberately depart from it after seeing it reappear in notation. In an ensemble setting, a shared display could also help musicians recall and respond to material introduced by one another rather than relying solely on memory. Conversely, performers might adapt their playing to the system—for example, by emphasizing downbeats to obtain a more stable transcription. These possible feedback loops could shape the direction of an improvisation even though the model itself remains reactive; we present them as design hypotheses rather than evaluated outcomes.

The current interface does not allow performers to approve, reject, or lock revisions. Its persistent note-level representation could support such controls in future versions, giving performers more direct authority over the generated score.

Limitations. Table 2 reports model inference latency, and the approximately 10 ms browser measurement covers online beat-tracking inference rather than the full round trip from MIDI input to visible notation. In the accompanying demonstration, display latency is barely noticeable during short performances. However, actual rendering time by OSMD can add over 100 ms for long performances, and screen refresh can take another 10 to 20 ms. These times could be reduced through incremental update techniques, and we leave them for future work.

In addition, the online beat tracker can produce noisy output because it operates with limited context and only on note-on events. Its provisional score is therefore sometimes unsuitable as a dependable performer-facing transcription, particularly during rests, sustained notes, or ambiguous meter. Improving online beat and downbeat tracking remains necessary before the display can reliably inform immediate performance decisions. Offline refinement improves recent measures but arrives several seconds later, so higher-quality notation necessarily lags behind the performance. This delay may limit the proposed solo and ensemble uses when musicians need to respond immediately.

5 Conclusion

We presented **Magic Quill**, a browser-based system that visualizes a live MIDI performance as an evolving engraved score. Its online model supports immediate provisional rendering, while asynchronous offline refinement revises earlier measures without losing correspondence to performed notes. Quantitative evaluation characterizes the trade-off between responsiveness and transcription accuracy, and the accompanying demonstration presents the system as an audience-facing visualization of machine interpretation in progress. Future work will improve the onset quantization modeling, incremental score rendering, and performer control over the generated score.

References

[1] David Kim-Boyle. Reframing the listening experience through the projected score. In *Proceedings of the International Conference on Technologies for Music Notation and Representation (TENOR)*, pages 181–185, 2018.

[2] Roger B Dannenberg. An on-line algorithm for real-time accompaniment. In *Proceedings of the International Computer Music Conference (ICMC)*, volume 84, pages 193–198, 1984.

[3] Bochen Li and Zhiyao Duan. An approach to score following for piano performances with the sustained effect. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 24(12): 2425–2438, 2016. doi: 10.1109/TASLP.2016.2611938.

[4] Christopher Raphael. Automated rhythm transcription. In *Proceedings of the 2nd International Symposium on Music Information Retrieval (ISMIR)*, October 2001.

[5] Andrea Cogliati, David Temperley, and Zhiyao Duan. Transcribing human piano performances into music notation. In *International Society for Music Information Retrieval Conference*, 2016.

[6] Eita Nakamura, Kazuyoshi Yoshii, and Shigeki Sagayama. Rhythm transcription of polyphonic piano music based on merged-output hmm for multiple voices. *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, 25(4):794–806, April 2017. ISSN 2329-9290. doi: 10.1109/TASLP.2017.2662479. URL `https://doi.org/10.1109/TASLP.2017.2662479`.

[7] Yuki Hiramatsu, Eita Nakamura, and Kazuyoshi Yoshii. Joint estimation of note values and voices for audio-to-score piano transcription. In *International Society for Music Information Retrieval Conference*, 2021.

[8] Lele Liu, Qiuqiang Kong, Veronica Morfi, and Emmanouil Benetos. Performance MIDI-to-score conversion by neural beat tracking. In *Proceedings of the 23rd International Society for Music Information Retrieval Conference (ISMIR)*, Bengaluru, India, December 2022.

[9] Tim Beyer and Angela Dai. End-to-end piano performance-MIDI to score conversion with transformers. In *International Society for Music Information Retrieval Conference*, 2024.

[10] Wei Zeng, Junchuan Zhao, and Ye Wang. Bridging piano transcription and rendering via disentangled score content and style. In *The Fourteenth International Conference on Learning Representations*, 2026.

[11] Apple Inc. Logic Pro, 2026. URL `https://www.apple.com/logic-pro/`. Version 11.0.

[12] Chris Donahue, Ian Simon, and Sander Dieleman. Piano genie. In *Proceedings of the 24th International Conference on Intelligent User Interfaces*, IUI '19, page 160–164, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450362726. doi: 10.1145/3301275.3302288.

[13] Yusong Wu, Tim Cooijmans, Kyle Kastner, Adam Roberts, Ian Simon, Alexander Scarlatos, Chris Donahue, Cassie Tarakajian, Shayegan Omidshafiei, Aaron Courville, Pablo Samuel Castro, Natasha Jaques, and Cheng-Zhi Anna Huang. Adaptive accompaniment with realchords. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

[14] Lancelot Blanchard, Perry Naseck, Stephen Brade, Kimaya Lecamwasam, Jordan Rudess, Cheng-Zhi Anna Huang, and Joseph Paradiso. The jam_bot, a real-time system for collaborative free improvisation with music language models. In *Proceedings of the 26th International Society for Music Information Retrieval Conference (ISMIR)*, 2025.

[15] OSMD Contributors and PhonicScore. Opensheetmusicdisplay: Open source musicxml renderer. `https://github.com/opensheetmusicdisplay/opensheetmusicdisplay`, 2026.

[16] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[17] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. Language models are few-shot learners. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pages 1877–1901, 2020.

[18] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomput.*, 568(C), February 2024. ISSN 0925-2312. doi: 10.1016/j.neucom.2023.127063.

[19] Carol L. Krumhansl. *Cognitive Foundations of Musical Pitch*. Oxford University Press, New York, 1990.

[20] Reiner Pope, Sholto Douglas, Aakanksha Chowdhery, Jacob Devlin, James Bradbury, Anselm Levskaya, Jonathan Heek, Kefan Xiao, Shivani Agrawal, and Jeff Dean. Efficiently scaling transformer inference. *arXiv [cs.LG]*, November 2022.

[21] Lele Liu, Veronica Morfi, and Emmanouil Benetos. Acpas: A dataset of aligned classical piano audio and scores for audio-to-score transcription. In *Proceedings of the 22nd International Society for Music Information Retrieval Conference (ISMIR 2021), Late-Breaking Demos*, 2021.

[22] Silvan David Peter, Carlos Eduardo Cancino-Chacón, Francesco Foscarin, Andrew Philip McLeod, Florian Henkel, Emmanouil Karystinaios, and Gerhard Widmer. Automatic note-level score-to-performance alignments in the asap dataset. *Transactions of the International Society for Music Information Retrieval (TISMIR)*, 5(1):20–36, 2022. doi: 10.5334/tismir.118.

[23] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

[24] C. Raffel, B. McFee, E. J. Humphrey, J. Salamon, O. Nieto, D. Liang, and D. P. W. Ellis. Mir_eval: A transparent implementation of common mir metrics. In *International Society for Music Information Retrieval Conference (ISMIR)*, 2014.

[25] Eita Nakamura, Emmanouil Benetos, Kazuyoshi Yoshii, and Simon Dixon. Towards complete polyphonic music transcription: Integrating multi-pitch detection and rhythm quantization. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 101–105, 2018. doi: 10.1109/ICASSP.2018.8461914.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction describe the system, its dual-timescale design, and the scope of the quantitative evaluation. Performer-facing applications are presented as hypotheses rather than evaluated outcomes.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The experiments discuss training-data bias, and Section 4 discusses online tracking errors, delayed offline refinement, and browser rendering costs together with their implications for audience- and performer-facing use.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: There are no theoretical results in this paper.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [No]

Justification: The paper discloses the datasets and splits, model architecture, training objective and schedule, evaluation metrics, and benchmark hardware. It does not provide every preprocessing detail, random seed, batch size, or executable evaluation pipeline needed for exact reproduction.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in

some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: A public web demonstration is provided, but the training code, model checkpoints, and reproduction scripts are not openly available with this submission. We plan to release them with documentation upon publication.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The method and experiments sections specify the datasets and evaluation splits, model inputs and architecture, losses, optimizer and learning rates, schedule, segment lengths, augmentation, training steps, evaluation tolerance, metrics, and inference hardware.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: The paper reports aggregate evaluation of one trained checkpoint on fixed test sets. It does not report repeated training runs, confidence intervals, or significance tests, so variability due to initialization and training is not quantified.

   Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [No]

   Justification: Inference benchmarks identify the MacBook M1 Pro used for measurement, but training hardware, memory, wall-clock time, and total project compute are not reported.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conforms to the NeurIPS Code of Ethics and does not involve human-subject experiments or collection of personal data.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [No]

Justification: The paper does not provide a dedicated analysis of both positive and negative societal impacts. It does discuss narrower risks relevant to intended use, including transcription errors, delayed feedback, and stylistic bias in the training data.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: The work does not release a high-risk dual-use model or a newly scraped dataset; the system operates on user-supplied symbolic MIDI performances.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [No]

Justification: The datasets and software libraries are credited through citations, but the paper does not explicitly enumerate the version and license of every external asset.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [No]

    Justification: The paper introduces a prototype, model, and web demonstration, but full code and model documentation are not included with the submission. Documentation is planned for the public release.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: The paper does not involve research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: LLMs are not an important, original, or non-standard component of the method or experiments.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.