RUMAA: Repeat-Aware Unified Music Audio Analysis for Score-Performance Alignment, Transcription, and Mistake Detection

Sungkyun Chang, Simon Dixon, Emmanouil Benetos

Centre for Digital Music, Queen Mary University of London, UK

***Abstract*—This study introduces RUMAA, a transformer-based framework for music performance analysis that unifies score-to-performance alignment, score-informed transcription, and mistake detection in a near end-to-end manner. Unlike prior methods addressing these tasks separately, RUMAA integrates them using pre-trained score and audio encoders and a novel tri-stream decoder capturing task interdependencies through proxy tasks. It aligns human-readable MusicXML scores with repeat symbols to full-length performance audio, overcoming traditional MIDI-based methods that rely on manually unfolded score-MIDI data with pre-specified repeat structures. RUMAA matches state-of-the-art alignment methods on non-repeated scores and outperforms them on scores with repeats in a public piano music dataset, while also delivering promising transcription and mistake detection results.**

1. INTRODUCTION

Music Performance Analysis [1] (MPA), a key area of Music Information Retrieval (MIR), investigates the relationship between a musical score and its performance across various genres and instruments. This study focuses on classical piano music, examining how performers adhere to the score, what deviations occur, and how audio translates into symbols. In classical music, where score fidelity is relatively critical, these questions support applications like tutoring [2] and assessment.

We address three essential tasks: *score-to-performance alignment*, *score-informed transcription*, and *mistake detection*. Scores, with symbolic repeats and expressions, contrast with audio performances. Prior work treated these tasks separately—e.g., alignment mapping notes, transcription converting audio, and mistake detection identifying errors [3], [4]—yet their interdependence is clear: alignment reveals matched or missing notes signaling mistakes, while transcription leverages alignment for accurate audio-to-symbol conversion.

This interdependence suggests a unified approach would be more effective. Additionally, conventional MIDI-based representations cannot handle repeat symbols, which are often ignored in practice but strictly followed in formal performances. This poses challenges for MIDI-based alignment methods requiring manually verified, repeat-unfolded scores. In contrast, RUMAA directly processes repeat symbols.

We propose RUMAA [ru:ma:] illustrated in Fig. 2, a transformer-based framework unifying these tasks with key features:

- *Proxy-driven multi-task model*: Employs three proxy tasks to encode target task interdependencies, enhancing multi-task learning for transcription, alignment, and mistake detection.
- *Multimodal and crossmodal seq-to-seq*: Utilizes pre-trained score and audio encoders to jointly process symbolic scores and audio, enabling crossmodal reasoning and producing structured outputs that reflect task dependencies.
- *Repeat handling*: Handles repeats by aligning to performed structure without pre-unfolded scores, enabling adaptation to real-world performance variations.

This unified approach enables robust and flexible music performance analysis, bridging symbolic and audio modalities through proxy-driven, crossmodal learning.

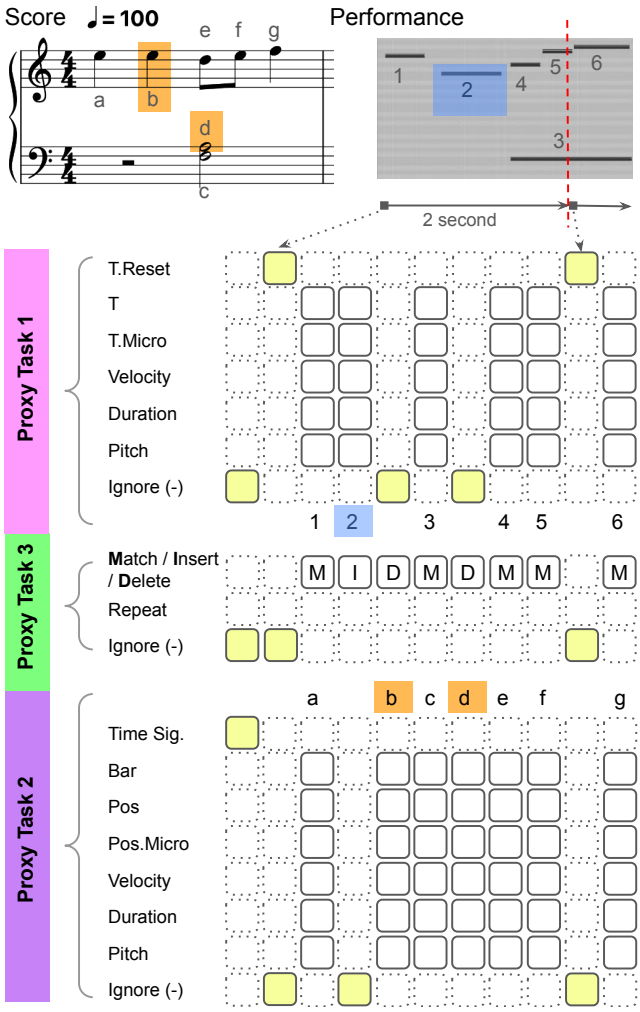

Fig. 1: **Token streams for proxy tasks. Top:** Example score and audio performance (visualized as a piano-roll), with orange highlights (missing) for missing score notes and pastel blue highlights (extra) for extra performance notes. **Bottom:** RUMAA decoder outputs for performance (Proxy Task 1), score (Proxy Task 2), and alignment (Proxy Task 3), as detailed in Section 3 and Table 1. Yellow boxes (□) denote exclusive tokens, and dotted boxes indicate silence tokens. Patterns ensure grayscale readability. Best viewed in color.

2. RELATED WORK

Score-performance alignment [5, pp. 115–166] employs cross-modal methods that directly align heterogeneous representations or symbolic methods that convert both modalities into a common space. Most methods adopt symbolic approaches, while cross-modal methods [6], [7] align in the audio domain using features like *chromagrams*, making note-level alignment challenging. Conventional DTW [8]–[10] and HMM-based [3] methods typically align transcribed symbolic sequences, incurring complexity and cascading errors from transcription.

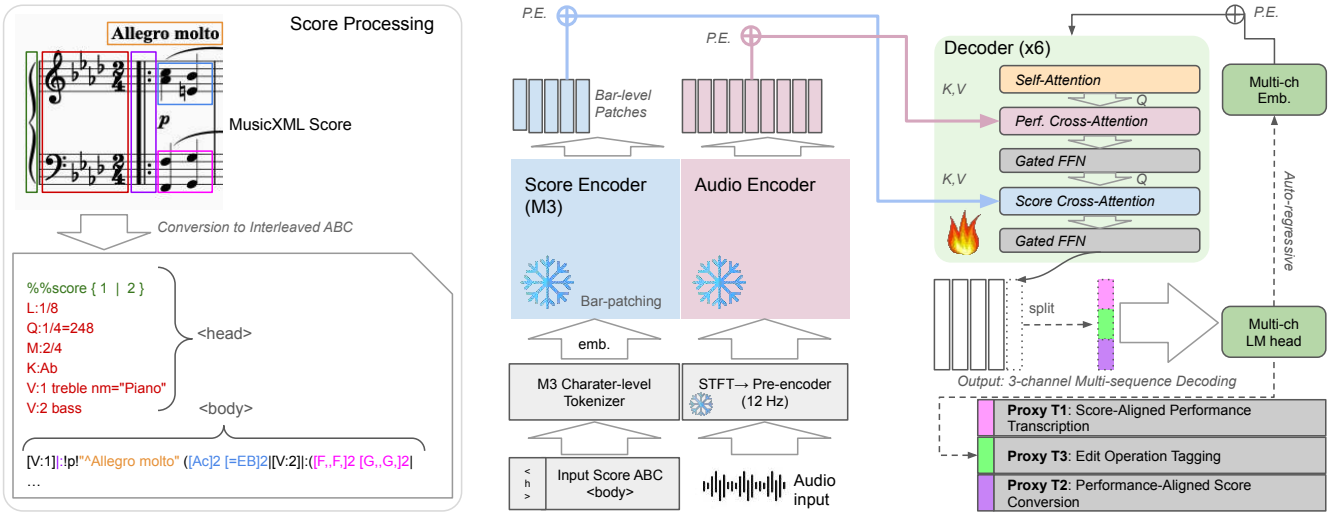

Fig. 2: **Overview of RUMAA framework**. **Left**: The input MusicXML score is converted into an interleaved ABC representation (Section 4.1). **Center**: The score and performance audio are encoded into score (Section 4.1) and audio embeddings (Section 4.2). **Right**: The decoder (Section 4.3), conditioned on score and audio context, autoregressively generates multi-channel tokens. Each channel corresponds to a proxy subtask (Section 3) related to our main tasks.

Repeat handling is a known issue [11] in alignment methods. MIDI-based approaches struggle since MIDI lacks explicit repeat notations. JumpDTW [11], [12] uniquely handles repeats by aligning image-domain scores with audio. Traditional methods rely on global structure analysis followed by local matching, while Transformer architectures may learn hierarchical patterns through multi-head attention, though long-context handling [13] remains challenging.

Recent deep learning methods include the GlueNote Transformer [14], which enhances DTW for symbolic alignment but requires external transcription. Chou et al. [15] proposed a Transformer for mistake detection, but we exclude it from baselines as it assumes pre-aligned 2-second score-audio pairs, making it unsuitable for inference on unaligned data.

Score-informed transcription methods [4], [16] demonstrate that joint modeling enables accurate transcription and simultaneous alignment with mistake detection, inspiring our unified transformer-based approach.

3. PROXY TASK DESIGN

We address three interdependent tasks: score-performance alignment, score-informed transcription, and mistake detection. To efficiently learn these task relationships and enable their joint inference, we formulate a proxy task that learns to generate a tokenized alignment sequence of performance (**T1**) and score (**T2**) events, including explicit edit operations (**T3**) through our tri-stream decoding.

The core idea (Figure 1) is to create a strict one-to-one alignment by explicitly modeling extra or missing notes. This sequence integrates edit operations such as <Match>, <Insert>, and <Delete>. When a note is present in the performance but absent in the score (an insertion) or vice versa (a deletion), the exclusive token <–> acts as a placeholder in the corresponding channel for the missing note. This maintains a consistent, perfectly aligned structure.

Figure 1 illustrates how each decoder channel corresponds to a proxy task. These tasks include:

- **T1.** **Score-Aligned Performance Transcription** outputs performance tokens with timing and note information.
- **T2.** **Performance-Aligned Score Conversion** outputs score tokens following performance order.
- **T3.** **Edit Operation Tagging** outputs alignment operation tokens.

Table 1: **Token definition.** Tokens marked with ★ are exclusive, silencing other multi-class tokens in their channel. <BOS> and <EOS> are global tokens that silence all channels. N includes a silence token per class.

Token	Range	N	Description
(Performance Tokens)			
T (onset time)	0:32	34	62.5 ms grid timing
T.Reset ★	None	2	Reset T every 2 seconds
T.Micro	-5:5	11	6.25 ms adjustment
Velocity	1:32	32	MIDI velocity, 4-unit steps
Duration	0:48	49	Up to 4s: 31.25/62.5/125 ms
Pitch	21:109	89	Piano MIDI pitch
- ★	None	2	Skip an extra note
(Score Tokens)			
Time Sig. ★	1/4 to 12/8	12	Time signatures (top/bottom)
Bar	0:50	51	Indexing bar
Pos	0:32	33	40-tick (32nd note) grid
Pos.Micro	-5:5	11	4-tick adjustment
Duration	0:48	49	40/80/160-tick steps
Pitch	21:109	89	Piano MIDI pitch
- ★	None	2	Skip a missing note
(Alignment Tokens)			
Insert/Delete/Match	None	4	Extra/Missing/Matched note
Repeat	None	2	Repeated notes or bar
- ★	None	2	Skip
BOS, EOS	None	2	Begin and end of sequence

This parallel decoding paradigm mirrors post-processed outputs from HMM/DTW-based methods using the Matchnote format [3], [10] adapted into a tokenized neural-friendly representation. While traditional approaches rely on global structure- or cluster-level alignment followed by local matching, our Transformer decoder directly learns these hierarchical patterns via cross-attention. Conditioning on full score input with barlines and repeats allows it to handle omissions and non-linear repeats without explicit rules.

Table 1 provides the complete token definitions. Our tri-stream tokenization (Table 1) extends CP-Words [17] by introducing structured tokens across three aligned streams: performance, score, and edit operations. Each note is encoded with onset, duration, and pitch tokens, including exclusive and silence tokens. Compared to serialized MIDI-like [18], [19] tokens, our design yields over 3× shorter sequences and enables simpler note-level alignment. By adopting two-level

quantization—<T> and <T.Micro> [20]—and a timing reset token [21], we achieve higher temporal resolution with fewer tokens than prior schemes, which typically require over 100 timing tokens per second. In contrast, our timing and duration representations are length-invariant, each using a fixed vocabulary of fewer than 50 tokens.

4. MODEL

Fig. 2 illustrates the structure of RUMAA, which integrates pre-trained score and audio encoders with a custom decoder. Our design specifically addresses the training memory bottleneck of long audio inputs—unlike symbolic alignment models [14] or short-audio transcription models (e.g., MT3 [18], [22]) without score. We adopt: (1) efficient score representation using ABC notation with bar-level patching, (2) optimized audio encoding to reduce token length, and (3) hierarchical cross-attention decoder with tri-stream output strategy. The following subsections detail these components.

4.1. Score Encoder

As shown in the left panel of Fig. 2, the score encoder converts MusicXML [23] to ABC notation [24], representing each bar with up to 64 characters [25]. Both notes and key musical elements (dynamics, repeats, keys, time signatures, and pedal) are retained.

We adopt the pre-trained M3 encoder from CLaMP2 [25], which relies on character-level tokenization and bar-level patching to process ABC notation. Each character is encoded as a 12-dim embedding, and 64 embeddings per bar are stacked into a single 768-dim bar-level token. A 12-block Transformer with multi-head self-attention, pre-trained to capture musical events without losing note-level details from long sequences, produces a 768-dim representation [25]. This output is linearly projected from 768 to 1,024 dimensions before being fed into the decoder described in Section 4.3.

4.2. Performance Audio Encoder

We use a spectrogram from 16 kHz mono audio (STFT: 2,048 samples window, 10ms hop). A pre-encoding layer (three ResNet blocks) processes features to produce 1024-dim outputs at 12 frames per second. Inspired by Music2Latent [26], this lower frame rate efficiently supports longer sequences without degrading transcription performance, as verified in preliminary experiments.

The audio encoder is a 12-layer self-attention Transformer, pre-trained following YourMT3+ [22] with three modifications: adapted for lower-frame-rate inputs, re-implemented with flash attention, and pre-trained decoder predicting MIDI velocity tokens (0-127). In RUMAA, we reuse only the pre-trained and frozen audio encoder, which produces 1,024-dim representations.

4.3. Decoder

The decoder, shown as the green box in the center of Fig. 2, is a 6-block Transformer autoregressively generating tri-stream outputs conditioned on audio (Section 4.2) and score (Section 4.1) features.

Our key architectural choice is hierarchical cross-attention, which involves separate conditioning for audio then score, in contrast to standard concatenation [15] or prepending input. Each block processes shifted outputs via self-attention, then cross-attention with audio, then score. This sequential approach resembles iterative transcription/score-following and is more efficient than simple context concatenation. Replacing hierarchical attention with simple concatenation leads to 1% performance drop despite similar model size.

Built on TorchScale [27], the Transformer uses a GatedFFN [28], [29] and extends standard decoder blocks with an additional cross-attention layer as mentioned above. The final 1024-dime latent representation is divided into three streams (1024 // 3), each processed by a multi-sequence language model (LM) head [17] to generate parallel sequences. Each head corresponds to a proxy task defined in Section 3: **T1**, **T2**, **T3**. The third head also has the potential to decode extended compound tokens, such as beat, tempo, or other attributes relevant to music performance modeling. Note that tokens with the same name across score and performance channels (e.g., <Pitch>; see Table 1) use separate embeddings and do not share weights.

5. EXPERIMENTAL SETUP

5.1. Data Preparation and Training

For pre-training the audio encoder, we use the YourMT3 dataset [22], combining 10 public multi-instrument transcription datasets including Maestro [30], reserving the official test split for evaluation. While prioritizing piano, we maintain multi-instrument pre-training, as initial findings show that mapping speech to singing and non-musical or percussive sounds to drums—or ignoring them—boosts robustness in live piano transcription.

For post-training the decoder, we use the (n)ASAP dataset [10], a Maestro [30] subset with 222 MusicXML [23] scores and 519 piano performances, including MIDI, audio, and manually verified note alignments [10]. We held out 20 movements and 50 recordings from six composers—Bach, Beethoven, Chopin, Haydn, Liszt, and Schubert—as the test set. MusicXML scores are converted to ABC-interleaved format [25] for the score encoder. Tri-stream tokens for the proxy tasks derived from (n)ASAP's alignment, MIDI, and score data. We randomly sample one-minute audio segments and extract up to 50 bars of score content that cover the corresponding passage.

To augment (n)ASAP's semi-professional performances, which limit mistake learning and rarely include repeats in 1-minute segments, we create five score-modulated versions, altering 10% of notes via pitch modulation (±5 semitons) or deletion, and five performance-modulated versions using Piano-SynMist [31] and MIDI-DDSP [32]. This expands the dataset tenfold for robust proxy task training. For repeat simulation, we add repeat symbols to random bars in 20% of ABC-interleaved scores lacking repeats, repeating the audio.

To evaluate score-performance alignment, we use the revised Vienna [33], with high-quality piano performances, transcriptions, MusicXML scores, and manual alignments. Score-informed transcription is tested on (n)ASAP [10], score-free transcription on Maestro [30]. Lastly, STPD [4] is used for mistake detection benchmark, with its score-MIDI converted to the ABC-interleaved format. All the evaluation datasets are isolated from the training dataset.

We trained our decoder alongside an off-the-shelf M3 [25] score encoder and a pre-trained performance audio encoder described in Section 4.1. For this post-training, we adopted AdamW-Scale optimizer [34] and scheduler from the prior work [22]. Training takes on three A6000 GPUs or H100 GPUs using a cosine schedule with initial and final learning rates of [1e-02, 1e-05] and a 1,000-step warm-up from 1e-03, spanning approximately two days.

5.2. Evaluation Metrics

Since our task includes transcription, we adopt a widely used ±50 ms onset tolerance [35] across all evaluations.

For score-to-performance alignment, we employ the note-level $\mathbf{F}_{\text{align}}$ metric [10]. This metric evaluates matched note pairs and inserted/deleted notes as *True Positives*, unmatched predicted notes as *False Positives*, and missing ground-truth notes as *False Negatives*. Originally designed for symbolic alignment tasks without repetitions, we adapt it by redefining repeated notes to be counted independently.

For score-informed transcription, we utilize two metrics derived from mir_eval [35]: $\mathbf{F}_{\text{on}}$ to evaluate note onset detection and $\mathbf{F}_{\text{off-vel}}$

to assess combined onset, offset, and velocity detection. Additionally, $\mathbf{MAE_{vel}}$ measures the mean absolute error of velocity (0–127).

For the mistake detection task in Table 4, we adopt four metrics consistent with prior studies [16], [36]: $\mathbf{F_{correct}}$, $\mathbf{Acc_{correct}}$ for correctly played notes, $\mathbf{F_{extra}}$, $\mathbf{Acc_{extra}}$ for erroneous notes not in the score, and $\mathbf{F_{missed}}$, $\mathbf{Acc_{missed}}$ for notes omitted from the performance despite being in the score.

6. RESULT

Table 2: **Note-level alignment (F_{align}) on the piano score-to-performance task.** "w/o repeat" indicates evaluation with all songs from Vienna [33] dataset, while "w/ repeat" indicates evaluation with two songs containing repeat symbols (*Mozart K331*, *Schubert D783*).

Model	w/o repeat	w/ repeat
(symbolic alignment)		
Nakamura HMM [3]	**99.0**	36.4
hDTW+sym [10]	98.5	28.2
GlueNote Transformer [14]	98.5	12.7
(symbolic-audio alignment)		
AMT [22] + Nakamura [3]	97.4	31.8
AMT [22] + hDTW+sym [10]	96.9	26.5
AMT [22] + GlueNote [14]	96.9	26.3
RUMAA (ours)	98.4	**98.4**

This section reports the evaluation of RUMAA on score-to-performance alignment, score-informed transcription, and mistake detection. Results are summarized in Table 2–4, showing its performance across these tasks within a unified framework, with particular focus on scores with repeats and score-informed processing.

6.1. Score-to-Performance Alignment

For the score-to-performance alignment task, Table 2 reports F_{align} values on the Vienna [33] piano dataset. The "without repeat" setting uses unfolded scores across all songs, while the "with repeat" setting applies original scores to songs with repeat symbols. The upper section, symbolic alignment, includes methods HMM [3], hDTW+sym [10], and GlueNote Transformer [14], aligning ground truth performance MIDI with unfolded score MIDI under ideal conditions without transcription errors nor repeats. Nakamura HMM achieves the highest performance on non-repeated scores, with RUMAA trailing by less than 1% (F1-score 98.4). On scores with repeats, however, these methods drop by up to 87%, while RUMAA maintains 98.4. Conventional methods [3], [10], [14] using score-MIDI cannot interpret repeat symbols, often aligning only the beginning and end of repeated scores. This is particularly noticeable in GlueNote, which struggles more with repeats.

The lower section, symbolic-audio alignment, reflects alignment errors that inherently include prior transcription errors, as the alignment is performed on transcribed symbolic data. On repeated scores, these methods drop by up to 87%, while RUMAA maintains 98.4. In symbolic-audio alignment, with transcription errors, baselines combining an external automatic music transcription (AMT) [22] with prior methods decline by up to 70% on repeats, but RUMAA outperforms them by over threefold, enhancing practicality without an external transcriber.

6.2. Score-informed Piano Transcription

Table 3 evaluates score-informed transcription on the (n)ASAP dataset [10], reporting onset F1, offset-velocity F1, and velocity MAE. Without score guidance on Maestro, RUMAA competes with top models like hFT-T [37] and YourMT3+ [22], though it falls short of IS-CRF [38] by about 2%. RUMAA's note offset-velocity prediction performance drops by up to 18% compared to other models. This may stem from inconsistent offset annotations in the multi-instrument dataset used to pre-train our audio encoder, unlike competing models optimized for Maestro. However, with score information on (n)ASAP, the proposed RUMAA achieves a near-perfect onset F1 score (99.1), surpassing all baselines by a clear margin, demonstrating that leveraging score information significantly enhances transcription performance.

Table 3: **Note-level transcription performance (Onset, Offset-Velocity F1 and Velocity MAE) on the score-informed piano transcription task**. "w/o score" indicates evaluation without score guidance, while "Score-informed" uses aligned score data from the (n)ASAP dataset [10] on the same test set. Models marked with an asterisk (*) are trained for multi-instrument transcription, while the others are piano-only.

Model	F_{on} ↑	$F_{off\text{-}vel}$ ↑	MAE_{vel} ↓
(w/o score: Maestro)			
hFT-T [37]	97.4	89.5	-
IS-CRF [38]	98.3	93.0	-
MT3* [18]	84.9	-	-
YourMT3+* [22]	97.0	-	-
RUMAA*	96.1	76.0	3.5
(w/o score: (n)ASAP)			
RUMAA*	95.9	75.8	4.6
(score-informed: (n)ASAP)			
RUMAA	**99.1**	**93.6**	**4.0**

Table 4: **Benchmark for piano mistake detection on the STPD [4] dataset.**

Model	w/o Score	Score-informed					
	F_{on}	$F_{correct}$	F_{extra}	F_{missed}	$A_{correct}$	A_{extra}	A_{missed}
Wang [16]	–	99.2	84.9	92.6	98.4	75.2	86.9
Ewert [36]	–	99.3	77.0	94.5	98.6	64.0	89.9
Benetos [4]	91.1	–	–	–	93.2	60.5	49.2
RUMAA	**92.8**	**99.5**	**89.2**	**95.3**	**98.7**	**80.3**	**90.4**

6.3. Piano Mistake Detection

For mistake detection, Table 4 presents a performance comparison on the STPD [4] piano dataset alongside previous NMF-based methods. RUMAA outperforms prior NMF-based approaches [4], [16], [36] in mistake detection. Without score guidance, it surpasses Benetos et al. [4] by approximately 2%; with score information, it achieves up to a 15% improvement in detecting extra and missed notes while maintaining top accuracy metrics.

7. DISCUSSION AND FUTURE WORK

RUMAA demonstrates that a unified model for score-performance alignment, transcription, and mistake detection achieves strong performance across all tasks. Notably, it performs well on scores with repeats, indicating effective modeling of MusicXML repeat structures. A key advantage of the unified audio and score encoders is reduced transcription error via score-informed decoding, in contrast to conventional AMT + HMM/DTW pipelines.

However, the model struggles with long audio sequences (over one minute) due to cross-chunk memory limits, restricting its use on extended real-world recordings [39] for alignment and mistake detection. Evaluations were also limited to relatively clean, single-instrument data, and online processing remains unexplored.

To address these limitations, future work could proceed along one of two avenues: developing memory-augmented architectures [40] to overcome sequence length constraints, or enhancing the model's generalizability by extending it to multi-instrument scores and more diverse, real-world recordings.

REFERENCES

[1] A. Lerch, C. Arthur, A. Pati, and S. Gururani, "Music performance analysis: A survey," in *International Society for Music Information Retrieval Conference (ISMIR)*, 2019.

[2] H. Zhang, V. Cheung, H. Nishioka, S. Dixon, and S. Furuya, "How does the teacher rate? observations from the neuropiano dataset," in *International Society for Music Information Retrieval Conference (ISMIR) – Late Breaking Demo*, 2024.

[3] E. Nakamura, K. Yoshii, and H. Katayose, "Performance error detection and post-processing for fast and accurate symbolic music alignment." in *International Society for Music Information Retrieval Conference (ISMIR)*, 2017.

[4] E. Benetos, A. Klapuri, and S. Dixon, "Score-informed transcription for automatic piano tutoring," in *IEEE European Signal Processing Conference (EUSIPCO)*, 2012, pp. 2153–2157.

[5] M. Müller, *Fundamentals of music processing: Audio, analysis, algorithms, applications*. Springer, 2015, vol. 5.

[6] M. Müller, A. Arzt, S. Balke, M. Dorfer, and G. Widmer, "Cross-modal music retrieval and applications: An overview of key methodologies," *IEEE Signal Processing Magazine*, vol. 36, no. 1, pp. 52–62, 2018.

[7] R. Agrawal, D. Wolff, and S. Dixon, "Structure-aware audio-to-score alignment using progressively dilated convolutional neural networks," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021.

[8] E. Keogh and C. A. Ratanamahatana, "Exact indexing of dynamic time warping," *Knowledge and information systems*, vol. 7, pp. 358–386, 2005.

[9] F. Simonetta, S. Ntalampiras, and F. Avanzini, "Audio-to-score alignment using deep automatic music transcription," in *2021 IEEE 23rd International Workshop on Multimedia Signal Processing (MMSP)*. IEEE, 2021, pp. 1–6.

[10] 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 (TISMIR)*, 2023.

[11] C. Fremerey, M. Müller, and M. Clausen, "Handling repeats and jumps in score-performance synchronization." in *International Society for Music Information Retrieval Conference (ISMIR)*, 2010.

[12] M. Shan and T. J. Tsai, "Improved handling of repeats and jumps in audio-sheet image synchronization," in *International Society for Music Information Retrieval Conference (ISMIR)*, 2020.

[13] Z. Dai, Z. Yang, Y. Yang, J. Carbonell, Q. V. Le, and R. Salakhutdinov, "Transformer-xl: Attentive language models beyond a fixed-length context," *arXiv preprint arXiv:1901.02860*, 2019.

[14] S. D. Peter and G. Widmer, "Thegluenote: Learned representations for robust and flexible note alignment," in *Proceedings of the 25th International Society for Music Information Retrieval Conference (ISMIR)*, 2024.

[15] B. S.-H. Chou, P. Jajal, N. J. Eliopoulos, T. Nadolsky, C.-Y. Yang, N. Ravi, J. C. Davis, K. Y.-J. Yun, and Y.-H. Lu, "Detecting music performance errors with transformers," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2025.

[16] S. Wang, S. Ewert, and S. Dixon, "Identifying missing and extra notes in piano recordings using score-informed dictionary learning," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 10, pp. 1877–1889, 2017.

[17] W.-Y. Hsiao, J.-Y. Liu, Y.-C. Yeh, and Y.-H. Yang, "Compound word transformer: Learning to compose full-song music over dynamic directed hypergraphs," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 1, 2021, pp. 178–186.

[18] J. Gardner, I. Simon, E. Manilow, C. Hawthorne, and J. Engel, "Mt3: Multi-task multitrack music transcription," in *International Conference on Learning Representations (ICLR)*, 2021.

[19] K. Bhandari, S. Chang, T. Lu, F. R. Enus, L. B. Bradshaw, D. Herremans, and S. Colton, "Improvnet: Generating controllable musical improvisations with iterative corruption refinement," *arXiv preprint arXiv:2502.04522*, 2025.

[20] J. Lenz and A. Mani, "Pertok: Expressive encoding and modeling of symbolic musical ideas and variations," in *International Society for Music Information Retrieval Conference (ISMIR)*, 2024.

[21] L. Bradshaw, H. Fan, A. Spangher, S. Biderman, and S. Colton, "Scaling self-supervised representation learning for symbolic piano performance," *arXiv preprint arXiv:2506.23869*, 2025.

[22] S. Chang, E. Benetos, H. Kirchhoff, and S. Dixon, "Yourmt3+: Multi-instrument music transcription with enhanced transformer architectures and cross-dataset stem augmentation," in *2024 IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, 2024.

[23] W. M. N. C. Group, "Musicxml 4.0," May 2021, accessed: 2025-03-27. [Online]. Available: https://www.w3.org/2021/06/musicxml40/

[24] C. Walshaw, "A statistical analysis of the abc music notation corpus: Exploring duplication," in *International Workshop on Folk Music Analysis*, A. Holzapfel, Ed., 2014.

[25] S. Wu, Y. Wang, R. Yuan, Z. Guo, X. Tan, G. Zhang, M. Zhou, J. Chen, X. Mu, Y. Gao *et al.*, "Clamp 2: Multimodal music information retrieval across 101 languages using large language models," *arXiv preprint arXiv:2410.13267*, 2024.

[26] M. Pasini, S. Lattner, and G. Fazekas, "Music2latent: Consistency autoencoders for latent audio compression," in *International Society for Music Information Retrieval Conference (ISMIR)*, 2024.

[27] S. Ma, H. Wang, S. Huang, W. Wang, Z. Chi, L. Dong, A. Benhaim, B. Patra, V. Chaudhary, X. Song, and F. Wei, "TorchScale: Transformers at scale," *CoRR*, vol. abs/2211.13184, 2022.

[28] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *Journal of machine learning research*, vol. 21, no. 140, pp. 1–67, 2020.

[29] J. Ding, S. Ma, L. Dong, X. Zhang, S. Huang, W. Wang, and F. Wei, "Longnet: Scaling transformers to 1,000,000,000 tokens," in *Proceedings of the 10th International Conference on Learning Representations*, 2023.

[30] C. Hawthorne and et al., "Enabling factorized piano music modeling and generation with the MAESTRO dataset," in *International Conference on Learning Representations (ICLR)*, 2019.

[31] A. Morsi, H. Zhang, A. Maezawa, S. Dixon, X. Serra *et al.*, "Simulating piano performance mistakes for music learning," in *Sound and Music Computing Conference*, 2024.

[32] Y. Wu, E. Manilow, Y. Deng, R. Swavely, K. Kastner, T. Cooijmans, A. Courville, C.-Z. A. Huang, and J. Engel, "Midi-ddsp: Detailed control of musical performance via hierarchical modeling," *arXiv preprint arXiv:2112.09312*, 2021.

[33] S. Gasser, M. Grachten, T. Gualtieri, and G. Widmer, "Vienna 4x22 Piano Corpus, Rematched," https://github.com/OFAI/vienna4x22_rematched, 2023, accessed: 2025-03-20.

[34] P. Nawrot, "Nanot5: Fast & simple pre-training and fine-tuning of t5 models with limited resources," in *Proceedings of the 3rd Workshop for Natural Language Processing Open Source Software (NLP-OSS 2023)*, 2023, pp. 95–101.

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

[36] S. Ewert, S. Wang, M. Müller, M. Sandler, S. Ewert, S. Wang, M. Muller *et al.*, "Score-informed identification of missing and extra notes in piano recordings," in *International Society for Music Information Retrieval Conference (ISMIR)*, 2016.

[37] K. Toyama, T. Akama, Y. Ikemiya, Y. Takida, W.-H. Liao, and Y. Mitsufuji, "Automatic piano transcription with hierarchical frequency-time transformer," in *International Society for Music Information Retrieval Conference (ISMIR)*, 2023.

[38] Y. Yan and Z. Duan, "Scoring time intervals using non-hierarchical transformer for automatic piano transcription," *arXiv preprint arXiv:2404.09466*, 2024.

[39] S. Flossmann, W. Goebl, M. Grachten, B. Niedermayer, and G. Widmer, "The magaloff project: An interim report," *Journal of New Music Research*, vol. 39, no. 4, pp. 363–377, 2010.

[40] A. Graves, G. Wayne, and I. Danihelka, "Neural turing machines," *arXiv preprint arXiv:1410.5401*, 2014.