STREAMING AUDIO-TO-SCORE TRANSCRIPTION

Anonymous authors
Paper under double-blind review

ABSTRACT

We present the first streaming music transcription system that converts live audio directly into human-readable sheet music. Our system is comprised of a cache-aware Conformer–RNN-T that emits a playback-ordered Intervals-and-Moments (InterMo) notation sequence in real-time, with an incremental sheet music rendering system for low latency. Our model emits InterMo tokens as solo piano, solo violin, and violin–piano duet performances unfold, directly producing time-aligned scores spanning up to three staves, without intermediate stages, updated at each musical moment. We benchmark streaming transcription on different latency configurations between 0.16 and 2.24 seconds, and show that our streaming model performs competitively with conventional offline audio-to-MIDI-to-score cascades. We additionally introduce a musically motivated "conductor" buffer, and show that a startup delay of only 5 seconds substantially increases streaming quality, especially in piano music. By bringing metrically structured, polyphonic notation into the streaming setting, our approach moves real-time music transcription toward human-facing applications in which live performances can be immediately captured, read, and communicated as conventional musical scores.

1 INTRODUCTION

Audio-to-score transcription aims to recover sheet music directly from performance audio, producing a representation from which musicians can read, follow, and perform. Unlike note-level automatic music transcription, this kind of transcription requires recovering not only timestamped pitch events, but also the metrical and notational structure needed to organize them into a human-readable score, including rhythm, simultaneity, voices, measures, and staves. Existing audio-to-score systems are formulated offline, producing a score only after the musical performance has been observed in full (Carvalho & Smaragdis, 2017; Zeng et al., 2024; Tamer et al., 2026).

Online music processing has instead focused on lower-level outputs, such as MIDI note events or beat positions, or on score following, where the reference score is supplied in advance (Wei et al., 2024; Meier et al., 2024; Park et al., 2025). Extending online transcription to the notation that musicians read would allow a score to take shape as the performance unfolds, much as streaming automatic speech recognition (ASR) produces written text as speech arrives (Andrusenko et al., 2026).

This paper introduces *streaming audio-to-score transcription*, in which a model incrementally generates musical notation from incoming audio with bounded acoustic lookahead. The central challenge is to recover metrical and polyphonic structure while the musical context is still unfolding. Waiting for more audio can help resolve ambiguity, but also delays the notation presented to the musician. A streaming system must therefore balance transcription quality with the timely production of readable notation.

Our approach builds on InterMo (Tamer et al., 2026), a playback-ordered representation of metric *intervals* and pitch-state-change *moments*, to connect streaming transcription with incremental score rendering. As shown in Figure 1a, a Conformer–RNN-T (Graves, 2012; Gulati et al., 2020)—a kind of RNN transducer—is trained without requiring timestamp supervision, marginalizing over monotonic paths that align audio to (generated) InterMo tokens. At inference time, as shown in Figure 1b, it emits this representation directly from incoming audio, serializing polyphonic events through multiple symbol emissions at a single acoustic step. Each completed Interval-and-Moment unit (M_0 through M_5 in the figure) can then be converted into notation and rendered without waiting for the rest of the performance. Together, the representation and transducer allow recognition and score construction to proceed incrementally.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

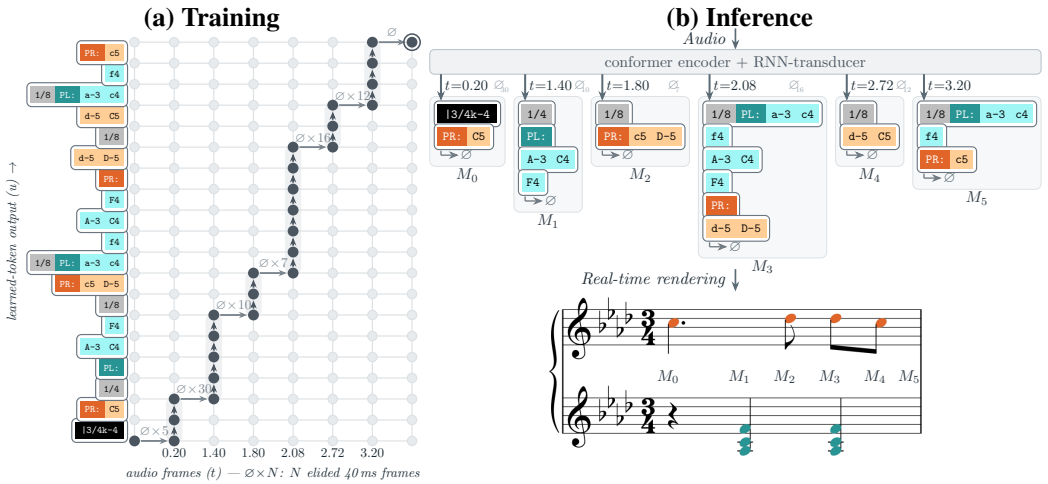

Figure 1: **From monotonic sequence training to real-time score rendering. (a)** Learning requires audio–score pairs, but no fine-grained alignment; the RNN-T training objective marginalizes over monotonic paths through an audio-frame/token lattice: blank transitions advance audio time and label transitions advance the score output. An example alignment path is highlighted. **(b)** As audio arrives, non-blank emissions are collected into Interval + Moment units (M_0–M_5) grounded in acoustic frames; the accumulated prefix is converted to notation and rendered while transcription continues. Both panels show the same musical excerpt. **(Not shown)** A natural by-product are timestamps grounding every rendered score element in the audio.

We evaluate the resulting system on solo piano, solo violin, and violin–piano duet performances across steady-state acoustic latencies from 0.16 to 2.24 s. A matched offline version of our model is competitive with or stronger than existing offline systems on violin and violin–piano transcription, while remaining behind the piano-specialist Rubato model on solo piano. With a 5 s startup-only "conductor" buffer, the streaming system approaches this matched offline reference as future context increases: at 2.24 s, its performance falls within the offline confidence interval on ASAP, the violin–piano duet, and Bach10. The buffer is especially important at low latency, where removing it produces widespread invalid outputs and higher error on piano and the duet. These results quantify the cost of producing structured, multi-staff notation online and show that much of the offline performance can be retained with bounded future context.

2 Related Work

Audio-to-score transcription. Most automatic music transcription systems consider recovering a performance-level pitch representation—such as note events or a piano roll—and meter inference, rhythm quantization, and other notation recovery as separate problems (Benetos et al., 2019; Liu et al., 2022; Beyer & Dai, 2024; Rouard et al., 2026). A separate line of work has studied direct audio-to-score transcription, mapping audio directly to symbolic notation with sequence-to-sequence or CTC-based models (Carvalho & Smaragdis, 2017; Román et al., 2018; 2019; Liu et al., 2021; Arroyo et al., 2022; Alfaro-Contreras et al., 2024; Zeng et al., 2024; Cummins et al., 2026). Several of these approaches are trained or evaluated on short musical excerpts whose boundaries are supplied externally (Román et al., 2018; 2019; Liu et al., 2021; Zeng et al., 2024). Two systems explicitly use bar/downbeat-aligned segmentation (Liu et al., 2021; Zeng et al., 2024). Extending such systems to unsegmented expressive performances therefore depends on reliable beat/downbeat tracking, which remains challenging for classical music with substantial tempo variation (Foscarin et al., 2024). Rubato (Tamer et al., 2026) removes the need for such metrical segmentation by introducing InterMo, whose locally self-contained representation allows predictions from independently decoded fixed-duration audio windows to be merged into a longer score. Our work builds on the same representation, but replaces independent window-level decoding with streaming transduction, incrementally emitting notation as audio arrives while restricting the model to bounded acoustic lookahead.

Online music processing. Producing human-facing notation from streaming audio, analogous to streaming ASR, has not been studied in music. The most similarly named task, but a fundamentally different one, is score following, which estimates the position of an ongoing performance within a reference score supplied in advance (Henkel & Widmer, 2021; Park et al., 2025). Other adjacent online music-processing tasks include real-time MIDI transcription, particularly for piano, which predicts note-on and note-off events without recovering a metrical grid (Dessein et al., 2010; Jeong, 2020; Wei et al., 2024; Kusaka & Maezawa, 2024), and online beat tracking, which estimates beat positions from an incoming performance without recovering the notes (Heydari & Duan, 2021; Meier et al., 2024). Streaming audio-to-score transcription must jointly recover metric and polyphonic structure and organize the resulting musical events into readable notation online.

Streaming sequence transduction. Alignment-free learning has long been attractive for music transcription, where precise frame-level annotations are expensive and often unavailable (Maman & Bermano, 2022; Weiß & Peeters, 2021). Prior work applied CTC to monophonic audio-to-score transcription and later introduced multi-label CTC (MCTC) to accommodate simultaneous pitches in multipitch estimation from weakly aligned audio–score pairs (Román et al., 2018; Weiß & Peeters, 2021). Yet, to our knowledge, RNN-T (Graves, 2012), whose emission mechanism is particularly well suited to serialized polyphony, has not previously been explored for music transcription.

RNN-T conditions each prediction on both the acoustic input and the previously emitted symbols, and may emit multiple symbols at the same acoustic step before producing a blank and advancing in time. Simultaneous musical events can therefore be serialized directly into a single output sequence without requiring a specialized multi-label objective or parallel prediction streams.

3 METHOD

Our method uses a Conformer–RNN-T to predict InterMo directly from audio. Streaming models restrict acoustic context with the cache-aware encoder described below, and decoded score prefixes are rendered incrementally. We omit explicit timestamp tokens; instead, RNN-T emission times provide audio grounding for decoded score elements.

3.1 TRANSDUCER LOSS FOR MULTI-STAFF POLYPHONIC NOTATION

An RNN transducer consists of an encoder, a predictor, and a joint network. The encoder maps the input audio to a sequence of acoustic representations, while the predictor maintains a hidden representation conditioned on the previously emitted symbols. At every acoustic position t and output position u, the joint network combines these representations and produces a distribution over the output vocabulary and the blank symbol, enabling one or multiple tokens from the same acoustic frame or moving to the next frame through blank.

Let $\mathbf{x} = (x_1, \ldots, x_T)$ denote the acoustic input and $\mathbf{y} = (y_1, \ldots, y_U)$ the target notation sequence; our training data consists of $(\mathbf{x}, \mathbf{y})$ pairs. An RNN transducer (Graves, 2012) models $P(\mathbf{y} \mid \mathbf{x})$ by marginalizing over monotonic alignments:

$$P(\mathbf{y} \mid \mathbf{x}) = \sum_{\boldsymbol{\pi} \in \mathcal{B}^{-1}(\mathbf{y})} P(\boldsymbol{\pi} \mid \mathbf{x}). \tag{1}$$

At a lattice position (t, u), emitting a blank advances the acoustic position t, while emitting a non-blank advances the output position u without advancing the encoder frame. Multiple output symbols can therefore be emitted at the same acoustic position.

Applying this loss to polyphonic notation requires a serialization whose output order remains monotonic with musical time. We use InterMo (Tamer et al., 2026), which represents a score as alternating metric intervals and moments of simultaneous pitch-state changes, with a deterministic ordering of events across staves. The resulting sequence advances monotonically with the performance while retaining the simultaneity and staff structure required to reconstruct notation.

While our model shares its score representation with the Rubato model, its tokenization and output space differ because of two properties of the transducer formulation. First, explicit timestamp symbols are unnecessary. During training, the transducer loss marginalizes over the possible placements of the

target symbols on the $T \times (U+1)$ alignment lattice, rather than requiring a timestamp for each target symbol. At inference, however, the alignment is no longer latent: every emitted non-blank symbol is associated with the encoder frame at which it was generated. We therefore obtain model-derived emission times for the score tokens without having to include timestamps in the target vocabulary or provide frame-level timestamp supervision.

Second, vocabulary size has a direct computational cost. At every lattice position, the joint network evaluates all V output symbols plus blank, giving a dense joint tensor of size

$$T \times (U+1) \times (V+1)$$

for each example before batching. Increasing the vocabulary therefore increases the memory and computation required by the transducer loss throughout the full alignment lattice. We consequently tokenize InterMo using a $V = 1800$ no-MIDI IntervalPiece vocabulary. As with subword units in speech recognition, this trades vocabulary size against target sequence length. This is particularly useful in the work reported here, because the same vocabulary must represent piano, violin, and multi-staff polyphonic notation.

3.2 Cache-Aware Streaming with a Conductor Buffer

We use the chunk-limited streaming Conformer formulation of Andrusenko et al. (2026), with cache-based inference following Noroozi et al. (2024). Its attention mask is parameterized by a left context L, a processing chunk C, and a right context R. At each streaming update, the encoder advances by C frames, retains up to L frames of cached history, and has access to R future frames beyond the current chunk. We refer to this right context as the *halo*: it is visible to the current update but is not part of the committed chunk.

The theoretical worst-case latency of a steady-state update is $C + R$ encoder frames (Andrusenko et al., 2026): a frame at the beginning of a chunk cannot be committed until the rest of the chunk and its halo are available, whereas a frame at the end waits only for the R-frame halo. For a fixed $C + R$ budget, allocating more frames to R and fewer to C improved recognition in the ASR experiments of Andrusenko et al. (2026), at the cost of more frequent encoder updates. In the limiting case $C = 1$, the encoder can commit at every frame, but this increases inference cost. We train with $C \in \{2, 4, 14, 26\}$ but, unless stated otherwise, fix $C = 4$ (160 ms) at inference, the smallest commit interval that runs in real time on a single Apple M4 CPU.

We make one music-specific modification to this speech-derived streaming schedule. In ensemble performance, a conductor's preparatory gesture gives performers a shared temporal reference before the first entrance. Score transcription has a related startup problem: meter, beat phase, and rhythmic grouping can only be weakly determined from the first local observations, even though these decisions affect how subsequent events are notated. We therefore introduce a one-time startup parameter B, which we call the *conductor buffer*.

We use $B = 5$ seconds (125 frames) throughout and do not tune the buffer duration.[1] Before the first output, the model accumulates B frames of audio. The final R frames form the usual halo, so the first update commits $B - R$ frames. Subsequent updates return to the standard schedule, committing C frames at a time while the available cached history grows until it reaches the left-context limit L. Figure 2 illustrates this schedule. Each red arrow denotes the span committed by one encoder update; the corresponding chunk is processed with the cached history and halo shown on the same row.

3.3 Sheet Music Rendering under Streaming Conditions

We modify the Verovio sheet music engraving library (Pugin et al., 2014) for streaming rendering. The target notation, InterMo, encodes duration implicitly: an uppercase pitch opens a note, a lowercase pitch closes it, and the metric intervals decoded in between determine how long it lasts. A note's notated value is therefore known only once its closing token has been decoded. Offline conversion sees both endpoints, but a streaming renderer must draw notes whose offsets have not yet arrived.

[1] We choose 5 seconds as a fixed heuristic motivated by beat-tracking practice, where the beginning of a recording is commonly treated as an initialization period; for example, `mir_eval` excludes beats before 5 seconds by default (Raffel et al., 2014).

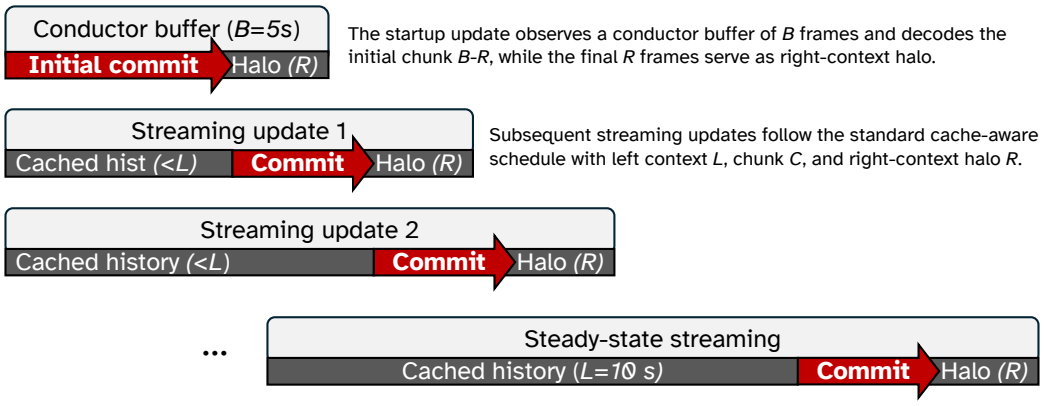

Figure 2: Cache-aware streaming with a conductor buffer. At startup, B frames are accumulated before the first update; $B - R$ are committed and the final R form the halo. Subsequent updates commit C frames at a time using the available cached history and an R-frame halo. Early updates have less than the full left context; in steady state, cached history is capped at L. Red arrows denote the span committed at one encoder update.

The displayed score is updated after every completed InterMo moment. Notes whose offsets have not yet arrived are rendered in gray with a provisional duration; when the corresponding offset is decoded, they are redrawn in black at their decoded duration. Provisional durations affect only the display and never modify the emitted InterMo sequence. At each update, the accumulated prefix is passed through the same notation conversion used offline, with provisional endpoints supplied for currently open notes. The score can therefore revise recent notation as additional output resolves previously incomplete events.

Naively re-engraving the complete score after every update would make rendering cost grow with performance length. Instead, we divide the score into musical systems with fixed bar ranges and cache each system's rendered SVG. After an update, Verovio is invoked only for the live system and for earlier systems whose notation has changed; unchanged systems are reused. Once an earlier system can no longer be affected by subsequent output, it is finalized and removed from further layout work. Thus, the settled portion of a growing score does not need to be repeatedly re-engraved. Details of the provisional-duration rule, score conversion, and rendering behavior are given in Appendix C.

4 Experimental Setup

4.1 Training Mix

We train on a mixture of real and score-conditioned synthetic audio spanning solo piano, solo violin, and violin–piano duet performances. Table 1 summarizes the resulting training corpus.

For real piano, we use aligned performances from the (n)ASAP collection (Foscarin et al., 2020; Peter et al., 2023). Real violin recordings come from Violin Etudes (Tamer et al., 2023). We augment both instruments substantially with score-conditioned synthesis.

For piano, we use the PDMX grand-staff score collection (Long et al., 2025) and the synthesis pipeline introduced by Tamer et al. (2026). Each score segment is rendered in two forms. The *PG* data are MIDI-timed DAW renderings, whereas the *PGE* data first pass through VirtuosoNet (Jeong et al., 2019) to introduce expressive timing and note-level deviations before audio synthesis with DawDreamer (Braun, 2021).

For violin–piano, we synthesize 6,008 public-domain MuseScore scores using two violin libraries and two piano banks under multiple room configurations. Instrument balance is varied by drawing the violin-to-piano SNR from $\mathcal{N}(0, 4)$ dB and clipping it to ± 12 dB. We additionally synthesize a collection of solo violin etudes using the same violin instruments.

Table 1: Composition of the training corpus. *Unique* hours are the audio actually covered (the union of segment intervals per recording); *seen* hours sum the segments presented to the model and therefore count overlapping windows again.

Collection	Instrumentation	Recordings	Utterances	Unique h	Seen h
Real recordings					
(n)ASAP	piano	245	11,406	17.0	79.3
Violin Etudes	violin	781	20,400	30.8	153.7
Subtotal		1,026	31,806	47.8	233.0
Synthesized from scores					
PDMX (PG)	piano	276,610	276,610	1,301.9	1,301.9
PDMX (PGE)	piano	244,185	244,185	1,139.1	1,139.1
MuseScore	violin+piano	23,710	174,455	963.6	1,323.2
Violin Etudes	violin	897	8,017	44.1	61.7
Subtotal		545,402	703,267	3,448.6	3,825.9
Total		546,428	735,073	3,496.5	4,058.8

To prevent score-level data leakage, we identify cross-collection overlap by content rather than filename.[2]

All examples use score-only InterMo targets. For the reported models, these are tokenized deterministically with a $V = 1800$ no-MIDI IntervalPiece vocabulary; acoustic timestamp tokens are not included in the target sequence. The RNN-T training requires audio to be cut into smaller segments since joint tensor size is bounded by both T (input audio, capped at 30 seconds) and U (output token length, each example containing at most 1,000 IntervalPieces.

4.2 Training Details

Experiments are conducted using NVIDIA NeMo 3.0. All models use the same 17-layer Conformer-L encoder ($d_{\text{model}} = 512$, 8 attention heads), a one-layer 640-dimensional LSTM predictor, and a 640-dimensional RNN-T joint network, for 118M parameters in total. We train three models: a matched offline reference, a streaming model without a conductor buffer ($B = 0$), and a streaming model with the 5 s conductor buffer ($B = 5$) described in Section 3.2. The two streaming models are trained independently, and each is decoded using the same startup condition used during training.

For both streaming models, we fix the left context to $L = 250$ frames and train over the Cartesian product

$$C \in \{2, 4, 14, 26\}, \qquad R \in \{0, 2, 4, 6, 10, 14, 26, 52\},$$

yielding 32 streaming geometries sampled uniformly during training. For the reported latency sweep, we fix $C = 4$ and evaluate all eight values of R above, so every reported streaming geometry is represented during training. With an encoder frame duration of 40 ms, these settings correspond to a worst-case steady-state acoustic latency of

$$(C + R) \times 40\,\text{ms},$$

ranging from 0.16 to 2.24 s. Streaming decoding is greedy, with at most 96 non-blank symbols emitted per encoder frame.

Both streaming models use the same held-out whole-performance validation set, comprising 100 solo-piano and 61 solo-violin recordings (11.75 h). The validation set contains no violin–piano duet recordings.

We train for 60,000 optimization steps, with 5,000 warmup steps to a peak learning rate of 5×10^{-4}, followed by cosine annealing. The effective batch size is 128 using gradient accumulation. Training uses bf16 mixed precision, while RNN-T joint logits are upcast to fp32 for loss computation.

[2]We fingerprint timestamp-stripped InterMo scores using 12-token shingles. Evaluation leakage is removed when directional containment is at least 0.5; within-corpus duplicate candidates are retained or removed only after a stricter whole-score comparison.

4.3 BASELINES

We compare against two strong offline systems that approach the task from opposite directions: a piano specialist that transcribes directly to score, and a large multi-instrument model that transcribes to MIDI and relies on a separate pipeline to produce notation.

Rubato (Tamer et al., 2026) is a 180M-parameter prompt-conditioned, offline piano transcription model that can perform MIDI or score transcription or beat tracking. It uses the same 100M-parameter Conformer-L encoder as our model but pairs it with a much larger Transformer decoder, and it is trained on far more real piano audio, including MIDI supervision from MAESTRO (Hawthorne et al., 2019). This makes it a strong piano-specialist offline baseline. We evaluate it with its time-aligned score-transcription prompt and beam size 4, decoding each audio window independently.

MuScriptor (Rouard et al., 2026) is a recent 1.3B-parameter offline transcription model for real-world music that covers a broad range of instruments, including piano and violin. It is fine-tuned on more than 11,000 hours of real recordings with symbolic note annotations aligned to the audio, obtained largely by synchronizing symbolic scores to the recordings, often via annotated bar-line positions. Thus, although MuScriptor outputs MIDI events rather than score notation, its real-audio supervision is itself score-derived and, where available, aligned using explicit metrical information. Together with its scale and instrument coverage, this makes it, to our knowledge, the strongest available multi-instrument baseline for our setting.

For sheet-music evaluation, we use MuScriptor's released score-transcription pipeline:[3] the 1.3B model transcribes audio to MIDI, Beat This (Foscarin et al., 2024) estimates the beat and downbeat grid used to quantize the predicted notes, and MuseScore converts the quantized MIDI into a score. We also evaluate two variants in which a learned MIDI-to-score model replaces both the quantization and score-construction stages, converting the predicted MIDI directly into grand-staff notation: M2ST (Beyer & Dai, 2024) and Joint APT-EPR (Zeng et al., 2026).[4]

4.4 EVALUATION SETS

All evaluation audio is from real performances. For piano, we use ATEPP (Zhang et al., 2022), with 1,495 recordings over 106 scores, and the ASAP test split, with 102 real recordings over 46 pieces. ATEPP provides the larger and more heterogeneous piano test, while ASAP gives a smaller set of aligned score–performance recordings.

For violin and violin–piano, we use three collections. KRAISLER (Kim et al., 2026) contains 20 recordings with isolated piano and violin stems as well as their duet; we use the dry condition and evaluate the mixture, while separate evaluations on piano and violin stems from this dataset are included in Appendix A. Bach10 (Duan et al., 2010) contributes 10 violin stems, and URMP (Li et al., 2019) contributes 34 violin stems from 21 ensemble pieces.

Bach10 and URMP do not provide corresponding MusicXML references. For these datasets, we construct the reference scores from the provided score-MIDI tracks using the same MuseScore 4 conversion used in MuScriptor's released score-transcription pipeline. Consequently, MuScriptor is evaluated against references produced by the same score-conversion system used to generate its predictions, giving it a favorable evaluation condition on these two datasets.

4.5 METRIC

We evaluate final notation with OMR-NED (Martinez-Sevilla et al., 2025). For a set of successfully converted scores, OMR-NED is

$$\text{OMR-NED} = \frac{\sum_i d(\hat{s}_i, s_i)}{\sum_i |s_i|}, \tag{2}$$

where d is the score edit distance and $|s_i|$ is the number of reference symbols. Lower is better. We report the metric as a percentage and give 95% bootstrap confidence intervals from 10,000 resamples.

[3] https://muscriptor.kyutai.org/

[4] We use only the transcription component of the joint system.

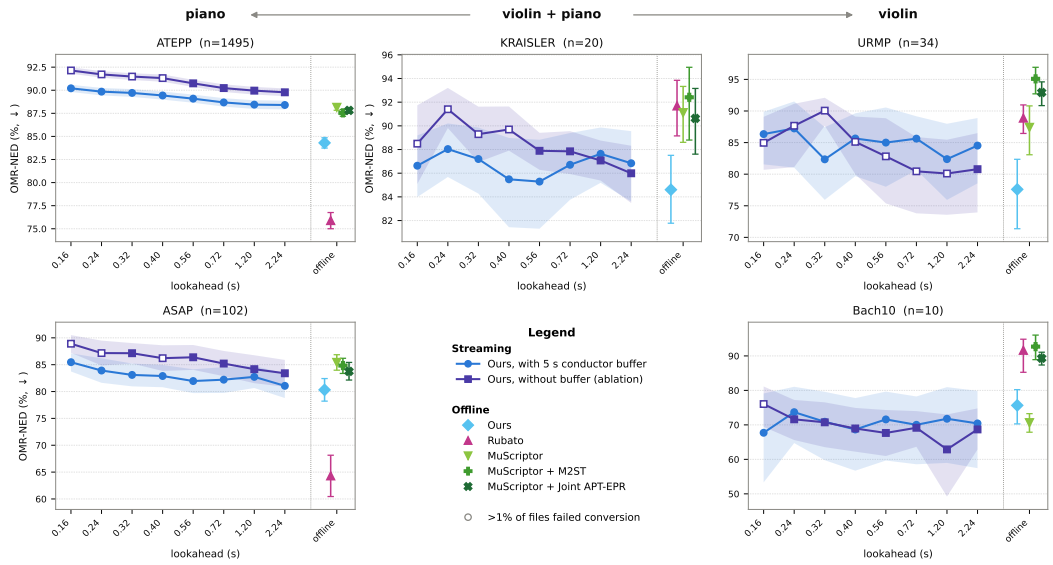

Figure 3: OMR-NED (lower is better) as a function of steady-state lookahead. Lines show our streaming model with the 5 s conductor buffer and, as an ablation, without it ($B = 0$). Points at the right of each panel show offline systems: an offline variant of our model trained on the same data, Rubato, MuScriptor, and two cascades that convert MuScriptor's MIDI output to a score with M2ST or with the MIDI-to-score component of Joint APT-EPR. Bands and error bars are 95% bootstrap confidence intervals; numerical values are given in Table 2. Hollow markers denote cells in which more than 1% of files failed score conversion (counts in Table 3). KRAISLER shows the violin–piano mixture. Panels use independent vertical scales.

A transcription that cannot be converted into a valid score has no OMR-NED value and is excluded from the ratio above. We therefore report the number of conversion failures (*fN*) alongside the affected results; in particular, a lower OMR-NED accompanied by more failures should not be interpreted as an improvement.

5 Results

Figure 3 gives the main quality–latency results. We first compare the offline configuration of our model with external systems, then the streaming system, and finally ablate the conductor buffer.

Offline comparison. Our offline model outperforms all other systems on the violin–piano duet and is competitive on solo violin: it is best on URMP ($n = 34$), while on Bach10 ($n = 10$) MuScriptor is better on average (within confidence intervals) when engraved with MuseScore, which is also used to engrave the Bach10 reference scores, but substantially worse when its MIDI is converted with M2ST or Joint APT-EPR. Our model's margin over the next-best system is about 6 OMR-NED points on the duet and about 10 points on URMP. On piano it is ahead of MuScriptor and both MIDI-to-score cascades by 3–5 points but behind Rubato, which was trained on a larger set of recorded piano performances. At least in the violin and duet settings, our architecture, although designed for streaming, is therefore not at a disadvantage relative to existing offline approaches.

Streaming with the conductor buffer. With the 5 s conductor buffer, the streaming system approaches its offline counterpart in most cases: at 2.24 s it is within the offline confidence interval on ASAP, the duet and Bach10, while gaps remain on ATEPP and URMP, which we analyse in Section B. On piano, additional lookahead gives a steady and significant improvement that levels off beyond about one second. This mirrors streaming speech recognition, where limited future context can recover part of the gap to full-context models (Zhang et al., 2020; Yu et al., 2021; Noroozi et al., 2024). On the duet and solo violin, by contrast, OMR-NED is largely flat across lookaheads.

Ablation: removing the conductor buffer. To isolate the effect of the conductor buffer, we train and decode the same model without it ($B = 0$). At low latency, this leads to substantial conversion failures (hollow markers in Figure 3; counts in Table 3): nearly all failed outputs (270 of 274) contain no bar lines at all, so no measure structure can be recovered. Where both conditions produce valid scores, removing the buffer increases OMR-NED on piano (significantly at nearly every lookahead) and on the duet at short lookahead, and makes no consistent difference on solo violin.

Error analysis. The quality–latency curves in Figure 3 do not behave uniformly across test sets: only on the two piano sets do the conductor buffer and additional lookahead give consistent, significant improvements. A sample-level analysis (Appendix B) traces part of this variation to the time signature. Our training data, rendered largely from public-domain scores, are dominated by common time (56% of segments in 4/4), whereas the share of 4/4 pieces varies widely across test sets: 20% in ATEPP, 32% in URMP, 42% in ASAP, 50% in KRAISLER and 80% in Bach10. When the early audio gives little metric evidence, the streaming model tends to fall back to 4/4 and rarely revises this choice later in the stream. On piano, simultaneous notes and bass lines make the metric grouping audible, so extra context can change the predicted meter; a single violin line gives little such evidence, and the prediction stays at 4/4 regardless of context. Accordingly, the gap between streaming and offline is largest on the sets furthest from common time (ATEPP, URMP) and absent on Bach10, where the 4/4 default is usually correct.

6 Conclusion and Limitations

We introduced streaming audio-to-score transcription, in which incoming performance audio is converted directly into incrementally renderable musical notation. By combining playback-ordered InterMo targets with RNN-T alignment, cache-aware acoustic context, and progressive score rendering, our system produces metrically structured, multi-staff notation while a performance is still unfolding. A matched offline version of the model establishes that the underlying architecture is competitive with strong offline systems on violin and violin–piano transcription, while the streaming experiments quantify the cost of restricting future acoustic context.

With a 5 s startup-only conductor buffer, the streaming system at the 2.24 s operating point falls within the matched offline confidence interval on ASAP, the violin–piano duet, and Bach10, with larger gaps remaining on ATEPP and URMP. Removing the conductor buffer substantially increases failures at low latency, indicating that startup context and steady-state lookahead play distinct roles in streaming transcription. Our error analysis further identifies recovery of metric structure, particularly time signature, as an important source of the remaining gap between streaming and offline transcription.

Our evaluation is limited to piano, violin, and violin–piano performances, and the solo-violin evaluation sets are relatively small, particularly Bach10 with 10 recordings. Generalization to other instruments, ensemble configurations, and musical styles therefore remains to be established. We also characterize streaming configurations using bounded acoustic context rather than systematically measured end-to-end wall-clock latency; actual latency additionally depends on feature extraction, decoding, score conversion, rendering, and hardware. Overall, our results show that human-readable, structured sheet music can be produced online while retaining much of the transcription quality of an otherwise matched full-context system.

AI use statement

We used generative AI tools to assist with method implementation and experimental infrastructure. Specifically, we used them to: (i) check the empirical equivalence of the implemented $[B, L, C, R]$ streaming-attention geometry with the conductor buffer and its behavior under streaming inference; (ii) assist with converting the trained model to an 8-bit quantized version for deployment on mobile devices; (iii) assist with the C++ implementation of the incremental notation-rendering system; and (iv) assist with experiment tracking and automation. AI-assisted code and implementations were reviewed and tested by the authors against the intended behavior and experimental outputs.

We did not use generative AI tools to formulate the central research question or hypotheses, develop the conceptual framework of streaming audio-to-score transcription, make the high-level methodological decisions of the work, generate or modify experimental results, or interpret the experimental

results. We did not use generative AI tools to formulate mathematical claims or proofs. Tasks involving synthetic dataset generation or qualitative/thematic data analysis are not applicable to this work.

Additionally, we used generative AI tools to improve the readability of the manuscript, search for and identify relevant literature, and assist with checking and formatting references. We have reviewed all AI-assisted work and take responsibility for the final content of this work, including text, claims, code, and other artifacts produced with the aid of generative AI.

REFERENCES

María Alfaro-Contreras, Antonio Ríos-Vila, Jose J. Valero-Mas, and Jorge Calvo-Zaragoza. A transformer approach for polyphonic audio-to-score transcription. In *2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 706–710, 2024. doi: 10.1109/ICASSP48485.2024.10447162.

Andrei Andrusenko, Vladimir Bataev, Lilit Grigoryan, Nune Tadevosyan, Vitaly Lavrukhin, and Boris Ginsburg. Reducing the offline-streaming gap for unified ASR transducer with consistency regularization. arXiv preprint arXiv:2604.19079, 2026.

Víctor Arroyo, Jose J. Valero-Mas, Jorge Calvo-Zaragoza, and Antonio Pertusa. Neural audio-to-score music transcription for unconstrained polyphony using compact output representations. In *2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4603–4607, 2022. doi: 10.1109/ICASSP43922.2022.9746239.

Emmanouil Benetos, Simon Dixon, Zhiyao Duan, and Sebastian Ewert. Automatic music transcription: An overview. *IEEE Signal Processing Magazine*, 36(1):20–30, 2019. doi: 10.1109/MSP.2018.2869928.

Tim Beyer and Angela Dai. End-to-end piano performance-MIDI to score conversion with transformers. In *Proceedings of the 25th International Society for Music Information Retrieval Conference (ISMIR)*, pp. 319–326, 2024. doi: 10.5281/zenodo.14877339.

David Braun. DawDreamer: Bridging the gap between digital audio workstations and python interfaces. Extended Abstracts for the Late-Breaking Demo Session of the 22nd International Society for Music Information Retrieval Conference, 2021.

Ralf Gunter Correa Carvalho and Paris Smaragdis. Towards end-to-end polyphonic music transcription: Transforming music audio directly to a score. In *2017 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pp. 151–155, 2017. doi: 10.1109/WASPAA.2017.8170013.

Eoin Cummins, Zhongyi Huang, Alexandre D'Hooge, Zhuoro Mo, and Yaolong Ju. Audio-to-score transcription using pre-trained features, data augmentation, and the new SheetSage-A2S dataset. In *Proceedings of the 34th ACM International Conference on Multimedia*, 2026. doi: 10.1145/3767308.3835653.

Arnaud Dessein, Arshia Cont, and Guillaume Lemaitre. Real-time polyphonic music transcription with non-negative matrix factorization and beta-divergence. In *Proceedings of the 11th International Society for Music Information Retrieval Conference (ISMIR)*, pp. 489–494, 2010.

Zhiyao Duan, Bryan Pardo, and Changshui Zhang. Multiple fundamental frequency estimation by modeling spectral peaks and non-peak regions. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(8):2121–2133, 2010. doi: 10.1109/TASL.2010.2042119.

Francesco Foscarin, Andrew McLeod, Philippe Rigaux, Florent Jacquemard, and Masahiko Sakai. ASAP: A dataset of aligned scores and performances for piano transcription. In *Proceedings of the 21st International Society for Music Information Retrieval Conference (ISMIR)*, pp. 534–541, 2020. doi: 10.5281/zenodo.4245490.

Francesco Foscarin, Jan Schlüter, and Gerhard Widmer. Beat this! Accurate beat tracking without DBN postprocessing. In *Proceedings of the 25th International Society for Music Information Retrieval Conference (ISMIR)*, pp. 962–969, 2024. doi: 10.5281/zenodo.14877491.

Alex Graves. Sequence transduction with recurrent neural networks. *arXiv preprint arXiv:1211.3711*, 2012. doi: 10.48550/arXiv.1211.3711.

Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, and Ruoming Pang. Conformer: Convolution-augmented transformer for speech recognition. In *Interspeech*, pp. 5036–5040, 2020. doi: 10.21437/Interspeech.2020-3015.

Curtis Hawthorne, Andriy Stasyuk, Adam Roberts, Ian Simon, Cheng-Zhi Anna Huang, Sander Dieleman, Erich Elsen, Jesse Engel, and Douglas Eck. Enabling Factorized Piano Music Modeling and Generation with the MAESTRO Dataset. In *International Conference on Learning Representations*, 2019. doi: 10.48550/arXiv.1810.12247.

Florian Henkel and Gerhard Widmer. Real-time music following in score sheet images via multi-resolution prediction. *Frontiers in Computer Science*, 3:718340, 2021. doi: 10.3389/fcomp.2021.718340.

Mojtaba Heydari and Zhiyao Duan. Don't look back: An online beat tracking method using RNN and enhanced particle filtering. In *2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 236–240, 2021. doi: 10.1109/ICASSP39728.2021.9413915.

Dasaem Jeong. Real-time automatic piano music transcription system. In *Late-Breaking/Demo Session of the 21st International Society for Music Information Retrieval Conference (ISMIR)*, pp. 4–6, 2020.

Dasaem Jeong, Taegyun Kwon, Yoojin Kim, Kyogu Lee, and Juhan Nam. VirtuosoNet: A hierarchical RNN-based system for modeling expressive piano performance. In *Proceedings of the 20th International Society for Music Information Retrieval Conference*, pp. 908–915, 2019. doi: 10.5281/zenodo.3527962.

Hyemi Kim, Jiyun Park, Sein Lee, Taegyun Kwon, Sunjae Won, and Juhan Nam. KRAISLER: A multi-track dataset of piano and violin duet recordings for music information retrieval research. *Transactions of the International Society for Music Information Retrieval*, 9(1):456–473, 2026. doi: 10.5334/tismir.338.

Yuta Kusaka and Akira Maezawa. Mobile-AMT: Real-time polyphonic piano transcription for in-the-wild recordings. In *32nd European Signal Processing Conference (EUSIPCO)*, pp. 36–40, 2024. doi: 10.23919/EUSIPCO63174.2024.10715008.

Bochen Li, Xinzhao Liu, Karthik Dinesh, Zhiyao Duan, and Gaurav Sharma. Creating a multitrack classical music performance dataset for multimodal music analysis: Challenges, insights, and applications. *IEEE Transactions on Multimedia*, 21(2):522–535, 2019. doi: 10.1109/TMM.2018.2856090.

Lele Liu, Veronica Morfi, and Emmanouil Benetos. Joint multi-pitch detection and score transcription for polyphonic piano music. In *2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 281–285, 2021. doi: 10.1109/ICASSP39728.2021.9413601.

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*, pp. 395–402, 2022. doi: 10.5281/zenodo.7316682.

Phillip Long, Zachary Novack, Taylor Berg-Kirkpatrick, and Julian McAuley. PDMX: A large-scale public domain MusicXML dataset for symbolic music processing. In *2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5, 2025. doi: 10.1109/ICASSP49660.2025.10890217.

Ben Maman and Amit H. Bermano. Unaligned supervision for automatic music transcription in the wild. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 14918–14934. PMLR, 2022.

Juan C. Martinez-Sevilla, Joan Cerveto-Serrano, Noelia Luna-Barahona, Greg Chapman, Craig Sapp, David Rizo, and Jorge Calvo-Zaragoza. Sheet music benchmark: Standardized optical music recognition evaluation. In *Proceedings of the 26th International Society for Music Information Retrieval Conference (ISMIR)*, pp. 618–625, Daejeon, South Korea, 2025. ISMIR. doi: 10.5281/zenodo.17811446.

Peter Meier, Ching-Yu Chiu, and Meinard Müller. A real-time beat tracking system with zero latency and enhanced controllability. *Transactions of the International Society for Music Information Retrieval*, 7(1):213–227, 2024. doi: 10.5334/tismir.189.

Vahid Noroozi, Somshubra Majumdar, Ankur Kumar, Jagadeesh Balam, and Boris Ginsburg. Stateful conformer with cache-based inference for streaming automatic speech recognition. In *2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 12041–12045, 2024. doi: 10.1109/ICASSP48485.2024.10446861.

Jiyun Park, Carlos Eduardo Cancino-Chacón, Suhit Chiruthapudi, and Juhan Nam. Matchmaker: An open-source library for real-time piano score following and systematic evaluation. In *Proceedings of the 26th International Society for Music Information Retrieval Conference*, pp. 91–99, Daejeon, South Korea, 2025. doi: 10.5281/zenodo.17811551.

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*, 6(1):27–42, 2023. doi: 10.5334/tismir.149.

Laurent Pugin, Rodolfo Zitellini, and Perry Roland. Verovio: A library for engraving MEI music notation into SVG. In *Proceedings of the 15th International Society for Music Information Retrieval Conference*, pp. 107–112, 2014.

Colin Raffel, Brian McFee, Eric J. Humphrey, Justin Salamon, Oriol Nieto, Dawen Liang, and Daniel P. W. Ellis. mir_eval: A transparent implementation of common MIR metrics. In *Proceedings of the 15th International Society for Music Information Retrieval Conference (ISMIR)*, pp. 367–372, Taipei, Taiwan, 2014. doi: 10.5281/zenodo.1416528.

Miguel A. Román, Antonio Pertusa, and Jorge Calvo-Zaragoza. An end-to-end framework for audio-to-score music transcription on monophonic excerpts. In *Proceedings of the 19th International Society for Music Information Retrieval Conference*, pp. 34–41, 2018. doi: 10.5281/zenodo.1492337.

Miguel A. Román, Antonio Pertusa, and Jorge Calvo-Zaragoza. A holistic approach to polyphonic music transcription with neural networks. In *Proceedings of the 20th International Society for Music Information Retrieval Conference*, pp. 731–737, 2019. doi: 10.5281/zenodo.3527914.

Simon Rouard, Michael Krause, Axel Roebel, Carl-Johann Simon-Gabriel, and Alexandre Défossez. MuScriptor: An open model for multi-instrument music transcription. arXiv preprint arXiv:2607.08168, 2026.

Nazif Can Tamer, Yiğitcan Özer, Meinard Müller, and Xavier Serra. High-resolution violin transcription using weak labels. In *Proceedings of the 24th International Society for Music Information Retrieval Conference (ISMIR)*, pp. 223–230, 2023. doi: 10.5281/zenodo.10265263.

Nazif Can Tamer, Victoria Ebert, Guang Yang, and Noah A. Smith. Rubato: Transcribing piano music with timestamps. *arXiv preprint arXiv:2605.24291*, 2026. doi: 10.48550/arXiv.2605.24291.

Weixing Wei, Jiahao Zhao, Yulun Wu, and Kazuyoshi Yoshii. Streaming piano transcription based on consistent onset and offset decoding with sustain pedal detection. In *Proceedings of the 25th International Society for Music Information Retrieval Conference (ISMIR)*, pp. 906–913, 2024. doi: 10.5281/zenodo.14877477.

Christof Weiß and Geoffroy Peeters. Learning multi-pitch estimation from weakly aligned score-audio pairs using a multi-label CTC loss. In *2021 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pp. 121–125, 2021. doi: 10.1109/WASPAA52581.2021.9632740.

Jiahui Yu, Wei Han, Anmol Gulati, Chung-Cheng Chiu, Bo Li, Tara N. Sainath, Yonghui Wu, and Ruoming Pang. Dual-mode ASR: Unify and improve streaming ASR with full-context modeling. In *International Conference on Learning Representations*, 2021. doi: 10.48550/arXiv.2010.06030.

Wei Zeng, Xian He, and Ye Wang. End-to-end real-world polyphonic piano audio-to-score transcription with hierarchical decoding. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, pp. 7788–7795, 2024. doi: 10.24963/ijcai.2024/862.

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

Huan Zhang, Jingjing Tang, Syed Rifat Mahmud Rafee, Simon Dixon, George Fazekas, and Geraint A Wiggins. ATEPP: A dataset of automatically transcribed expressive piano performance. In *Proceedings of the 23rd International Society for Music Information Retrieval Conference (ISMIR)*, pp. 446–453, Bengaluru, India, 2022. doi: 10.5281/zenodo.7342764.

Qian Zhang, Han Lu, Hasim Sak, Anshuman Tripathi, Erik McDermott, Stephen Koo, and Shankar Kumar. Transformer transducer: A streamable speech recognition model with transformer encoders and RNN-T loss. In *2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7829–7833, 2020. doi: 10.1109/ICASSP40776.2020.9053896.

A Additional Results

A.1 Complete OMR-NED values

Table 2 gives the numerical OMR-NED values behind Figure 3, for every system and lookahead, together with the isolated KRAISLER piano and violin stems. Values are micro-averaged over the files of each set (lower is better), with the half-width of the 95% bootstrap confidence interval. Files whose output cannot be converted into a valid score are excluded from the average; cells in which this affects more than 1% of the files are marked †, corresponding to the hollow markers in the figure, and Table 3 lists the number of failed files for every cell.

B Time-signature analysis

For each file we take the dominant time signature (the most frequent over bars) of the reference score and of each system's output. Table 4 reports how often the data and each system are in 4/4, meter accuracy, and how often a wrong first meter is corrected later in the piece. Streaming ranges are over the eight trained lookaheads; recovery is measured at 0.16, 0.56 and 2.24 s.

The model defaults to 4/4 with limited acoustic evidence. Across all sets, the streaming model writes 4/4 far more often than the references contain it, and more often than its training data would suggest (56% of training segments are in 4/4). With the conductor buffer, it writes 4/4 for about 80% of ATEPP pieces and for every solo-violin piece, so its meter accuracy is close to that of a trivial always-4/4 predictor (first row of Table 4). As we show next, this default is strongest where the audio carries the least metric information.

Context helps where the meter is audible. On piano, the conductor buffer changes this behaviour: on ASAP it raises meter accuracy from 31–42% to 46–50% at every lookahead, alongside its OMR-NED improvement. On solo violin it does not. With the buffer, the model writes 4/4 for every violin piece regardless of lookahead; without it, short-lookahead output often lacks bar lines altogether, and from 0.72 s on it too defaults to 4/4. We attribute the difference to the signal: simultaneous notes and bass lines mark the metric grouping in piano music, whereas a single melodic line carries much weaker metric cues.

The initial choice is rarely revised. The streaming model fixes the time signature at the first bar line. Of the pieces whose first predicted meter is wrong, fewer than 10% are corrected later in the stream. The training data, largely rendered from public-domain scores, offer little opportunity to learn such revisions: only 5.6% of training segments contain a time-signature change (1.3% for solo

	piano		KRAISLER			violin	
system	ATEPP	ASAP	piano	mix	violin	URMP	Bach10
n	1495	102	20	20	20	34	10
Offline							
Ours	84.3±0.6	80.3±2.1	75.8±7.0	84.6±2.9	83.6±5.2	77.6±5.5	75.7±5.0
Rubato	75.9±0.9	64.3±3.8	71.4±7.5	91.7±2.3	95.5±1.1	88.9±2.3	91.6±4.8
MuScriptor	88.1±0.3	85.4±1.4	82.6±5.4	91.1±2.4	86.2±4.6	87.4±3.9	70.6±2.7
MuScriptor + M2ST	87.6±0.4	84.8±1.4*	84.8±5.3	92.4±3.1	95.1±1.6	95.1±2.1	92.7±3.6
MuScriptor + Joint APT-EPR	87.8±0.3	83.8±1.6*	83.3±4.9	90.6±2.8	94.0±1.3	92.9±1.9	89.3±1.9
Ours, with 5 s conductor buffer							
0.16 s	90.2±0.4	85.5±1.7	84.0±4.1	86.6±2.6	89.4±2.5	86.3±4.2	67.7±12.9
0.24 s	89.8±0.4	83.9±2.3	84.6±3.5	88.0±2.3	88.5±2.2	87.2±5.2	73.7±8.2
0.32 s	89.7±0.4	83.1±2.1	86.4±2.6	87.2±2.8	87.9±2.8	82.4±5.7	70.9±9.9
0.40 s	89.4±0.4	82.9±2.1	83.0±4.3	85.5±3.8	86.7±4.0	85.6±5.0	68.7±10.5
0.56 s	89.1±0.4	82.0±2.2	84.0±4.1	85.3±3.8	88.6±2.1	85.0±6.3	71.6±10.0
0.72 s	88.7±0.5	82.2±2.4	84.8±3.8	86.7±2.8	86.8±3.9	85.6±4.3	70.0±9.8
1.20 s	88.4±0.5	82.7±2.1	84.4±4.2	87.6±2.3	88.1±1.6	82.4±6.0	71.8±11.0
2.24 s	88.4±0.5	81.1±2.3	79.9±4.3	86.8±3.0	89.1±1.9	84.5±5.2	70.4±11.2
Ours, without buffer (ablation, $B = 0$)							
0.16 s	92.1±0.3†	88.9±1.6†	84.3±4.6†	88.5±3.3†	91.0±1.6†	85.0±4.2†	76.0±5.9†
0.24 s	91.7±0.4†	87.2±2.3†	88.8±2.4†	91.4±1.7†	90.4±2.9†	87.7±5.0†	71.6±5.8
0.32 s	91.5±0.4†	87.1±1.9	87.0±3.6†	89.3±2.3†	89.1±2.1†	90.1±2.2†	70.7±6.5
0.40 s	91.3±0.4†	86.2±2.4†	85.4±3.9†	89.7±1.9†	88.2±2.6†	85.1±4.6†	69.0±6.3
0.56 s	90.7±0.4	86.4±2.3	83.2±4.9†	87.9±1.6	86.9±4.9†	82.8±6.7†	67.7±6.7
0.72 s	90.2±0.4	85.2±2.4	84.6±4.3	87.8±1.8	84.5±5.4	80.5±6.0†	69.2±5.2
1.20 s	90.0±0.4	84.2±2.6	83.1±4.3	87.1±1.7	87.7±2.2	80.1±5.9†	62.9±11.9
2.24 s	89.8±0.4	83.4±2.5	80.9±5.2	86.0±2.4	86.1±3.2	80.8±6.3	68.7±6.0

Table 2: Full OMR-NED results (micro-averaged, %, lower is better; ± half-width of the 95% bootstrap confidence interval). †More than 1% of files failed score conversion (hollow markers in Figure 3); failed files are excluded from the average, and counts are given in Table 3. The main text reports KRAISLER as the violin–piano mixture; the isolated piano and violin stems are included here for comparison. *The MIDI-to-score models of the M2ST and Joint APT-EPR cascades were trained on ASAP data that overlaps this test set.

violin). Rubato, although also trained largely on public-domain (PDMX) scores, behaves differently because it decodes chunks independently: a wrong meter in one chunk can be replaced in the next, and it recovers 73–79% of such pieces on the piano sets.

Consequences for the streaming–offline gap. The offline model, which sees the whole piece, recovers more non-4/4 meters, and this accounts for part of its advantage. On files where only the offline model predicts the correct meter, the streaming model's OMR-NED is higher by 8.4 points on ATEPP and 3.3 on ASAP, compared with about 3 and 0.7 on the remaining files. The meter does not explain the whole gap: on ATEPP, files on which both models get the meter wrong still differ by 3.5 points. Bach10 shows the reverse case. Eight of its ten pieces are in 4/4 and none changes meter, so the 4/4 default is usually correct, and the streaming model performs at the offline level (70.4 vs. 75.7; the paired difference is not significant).

C Streaming Conversion and Rendering

This appendix gives additional details on how the decoded InterMo prefix is turned into a progressively rendered score. Streaming and offline transcription use the same notation-conversion procedure; streaming differs only in that the current prefix may contain notes whose offsets have not yet been emitted and in that the rendered score is updated incrementally.

Converting a partial InterMo sequence. InterMo represents a note with an uppercase onset token and a matching lowercase offset token, with the metric intervals between them determining its duration. During streaming, an onset may therefore be available before its final notated duration is known. For display, we temporarily close such an open note using the duration of the most recently closed note on the same staff or, if none is available, the most recent metric interval. The provisional endpoint is constrained to extend beyond the current musical position and not past the end of the current bar.

		piano		KRAISLER			violin	
condition	lookahead	ATEPP	ASAP	piano	mix	violin	URMP	Bach10
n		1495	102	20	20	20	34	10
Ours, offline	–	1	0	0	0	0	0	0
5 s buffer	0.16 s	0	0	0	0	0	0	0
	0.24 s	0	0	0	0	0	0	0
	0.32 s	0	0	0	0	0	0	0
	0.40 s	0	0	0	0	0	0	0
	0.56 s	1	0	0	0	0	0	0
	0.72 s	1	0	0	0	0	0	0
	1.20 s	0	0	0	0	0	0	0
	2.24 s	0	0	0	0	0	0	0
$B = 0$	0.16 s	**24**	**6**	**3**	**2**	**12**	**18**	**1**
	0.24 s	**17**	**4**	**3**	**1**	**8**	**20**	0
	0.32 s	**19**	0	**4**	**1**	**11**	**14**	0
	0.40 s	**21**	**2**	**3**	**1**	**10**	**16**	0
	0.56 s	7	0	**2**	0	**9**	**17**	0
	0.72 s	6	0	0	0	0	**1**	0
	1.20 s	5	0	0	0	0	**1**	0
	2.24 s	5	0	0	0	0	0	0

Table 3: Files that failed score conversion. Bold: more than 1% of the set, shown as a hollow marker in Figure 3 and marked † in Table 2. The main text reports KRAISLER as the violin–piano mixture; the isolated piano and violin stems are included here for comparison.

	ATEPP	URMP	ASAP	KRAISLER	Bach10
reference in 4/4	20%	32%	42%	50%	80%
predicted 4/4					
Streaming (ours), $B = 0$	75–94%	24–100%	69–94%	65–100%	90–100%
Streaming (ours), 5 s buffer	79–80%	100%	73–75%	75–80%	100%
Offline (ours)	48%	88%	49%	25%	90%
Rubato	20%	62%	32%	25%	40%
meter accuracy					
Streaming (ours), $B = 0$	16–19%	3–32%	31–42%	30–50%	80%
Streaming (ours), 5 s buffer	18–19%	32%	46–50%	40%	80%
Offline (ours)	25%	38%	47%	20%	70%
Rubato	32%	26%	52%	30%	30%
wrong first meter corrected later					
Streaming (ours), $B = 0$	2–5%	0%	5–13%	0%	0/2
Streaming (ours), 5 s buffer	3–4%	0%	4–8%	0%	0/2
Offline (ours)	2%	0%	6%	0%	0/3
Rubato	73%	48%	79%	18%	3/6

Table 4: Time signatures in the reference and in system outputs. An always-4/4 guesser has meter accuracy equal to the first row.

Provisional notes are rendered in gray and redrawn in black at their decoded duration once the corresponding offset arrives. For example, if the model has emitted an onset for D4 but has not yet emitted its offset, D4 is displayed with a provisional duration; when the subsequent InterMo tokens establish that the note lasts a quarter note, the same note is redrawn in black as a quarter note. These provisional values are used only for visualization and never alter the emitted token sequence.

After every completed InterMo moment, the accumulated prefix is converted to notation exactly as in offline transcription, with provisional endpoints supplied for any remaining open notes. Conversion pairs note onsets and offsets, assigns the resulting notes to voices and clefs, and produces conventional notated measures. Because the entire decoded prefix is reconsidered, recent notation may change when new tokens resolve an open note or otherwise change the preferred conversion. After supplying

provisional endpoints for open notes, we verified that streaming conversion matched offline conversion of the same prefix at every update of 180 transcriptions (53,574 updates).

Incremental engraving. Rendering the entire accumulated score after every update would make engraving cost increase with performance length. We therefore divide the score into musical systems with fixed bar ranges and render each system as a separate SVG using Verovio (Pugin et al., 2014). After each update, the live system is re-engraved, together with any earlier system whose notation has changed; unchanged systems reuse their cached rendering.

Earlier systems may still change while they contain unresolved notation, for example because a note remains open or the conversion of recent material has not yet stabilized. Once a system can no longer be affected by subsequent output, it is marked final and excluded from further layout work. Across the same 180 transcriptions, approximately 1.05 systems were re-engraved per update on average. Thus rendering remains concentrated near the live edge of the score rather than repeatedly processing the full performance.