QUANTIZING RHYTHM WITH MUSIC FOUNDATION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Rhythm quantization, mapping the expressive timings of a musical performance into a score's precise note values, must be learned from fewer than a thousand paired performances and fewer than two hundred scores. We show that a pretrained symbolic-music foundation model can be adapted into a rhythm quantizer with no new parameters: fed the performance as its anticipated control stream, the Anticipatory Music Transformer becomes a fixed-lookahead transcriber of entire songs, fine-tuned on interleaved pairs of events from a performance and its corresponding score. Musical pretraining substantially improves this adaptation. Under an identical training recipe and checkpoint-selection rule, the pretrained model more than doubles whole-song strict note F1 relative to random initialization (8.7% vs. 3.2%). We further find that teacher-forced fine-tuning in this data-limited regime is susceptible to compounding errors during whole-song autoregressive inference. To address this, we introduce a three-stage training curriculum that progressively bridges the distribution shift between teacher-forced training and model-generated inference. On the 59-performance ASAP test set, our model reaches 9.33% strict note F1 and 77.93% inter-onset-interval F1, compared with 8.96% and 67.81%, respectively, for the strongest task-specific baseline. These gains are accompanied by a substantially higher rate of producing outputs at the reference score's time scale.

1 INTRODUCTION

A musician never plays exactly what is written. Rubato stretches the beat, chords spread, and expressive timing can blur the rhythmic structure encoded by the score (Desain & Honing, 1989; Cemgil et al., 2000; Nakamura & Yoshii, 2021). Rhythm quantization, which maps the expressive timings of a performance onto the beat-aligned grid of a score (Nakamura et al., 2018), therefore requires a strong prior over plausible musical scores. Yet the paired data available for learning this mapping are limited: in the benchmark split used by the strongest recent systems (Beyer & Dai, 2024; Zeng et al., 2026), ASAP (Foscarin et al., 2020) provides 822 training performances of only 176 works.

Symbolic-music foundation models pretrained on large corpora carry such a prior (Huang et al., 2019; Zeng et al., 2021; Thickstun et al., 2024; Qu et al., 2025), but they are designed to generate music rather than transcribe it. We show how to transfer this prior to transcription without adding task-specific parameters. The Anticipatory Music Transformer (AMT; Thickstun et al., 2024), a 360M-parameter decoder pretrained on Lakh MIDI, interleaves the events it generates with *control* events drawn from a bounded future. Supplying the performance as the control stream turns AMT into a transducer with fixed lookahead, analogous to simultaneous translation under a wait-k policy (Ma et al., 2019; Koshkin et al., 2024): each score note is generated after the model has seen the corresponding performance note and a bounded number of subsequent notes. We call the resulting model AMT-Q; it predicts each score note's onset and duration in beats (Section 3).

Whole-song autoregressive decoding is a central difficulty: a piece can span thousands of notes, so each decoding window depends partly on earlier predictions. Teacher-forced fine-tuning performs poorly as these generated contexts drift out of the meager training distribution. We address this with score-token dropout and checkpoint selection using validation rollouts, followed by on-policy DAgger training (Ross et al., 2011) (Figure 1).

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.

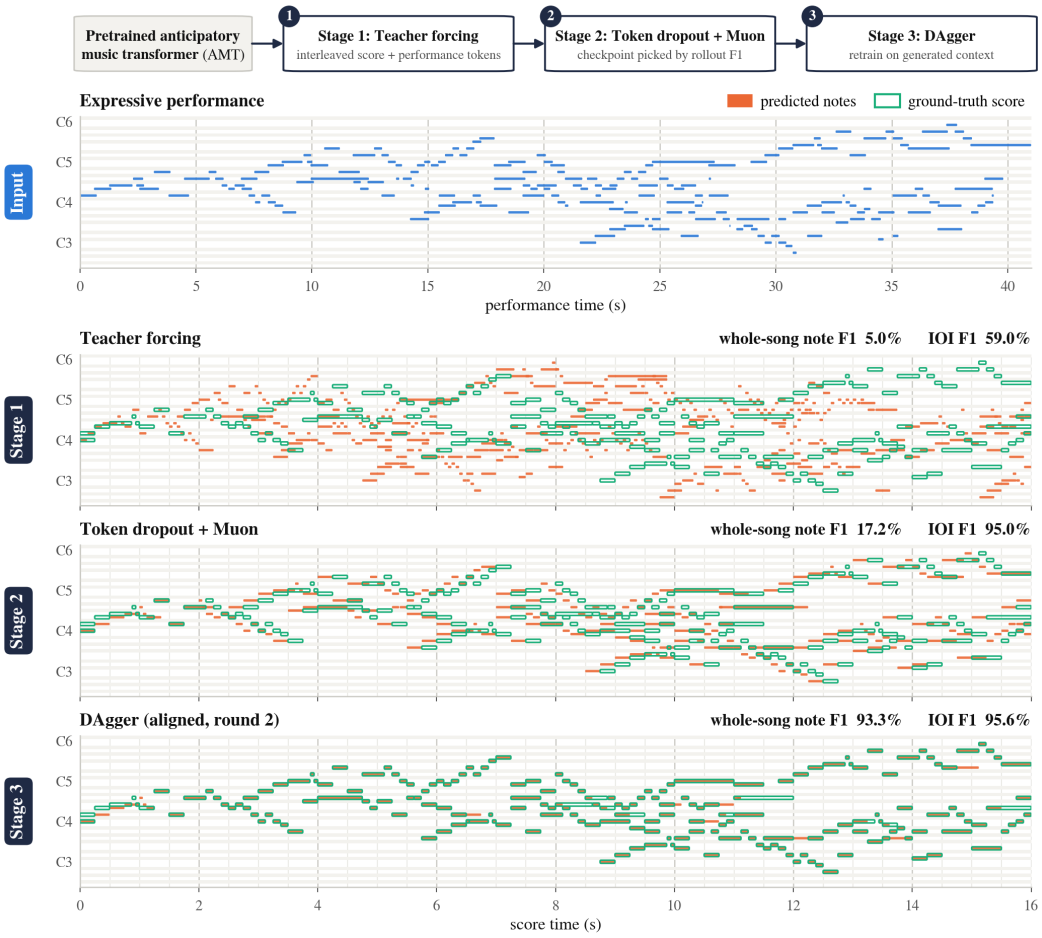


Figure 1: **Three-stage training curriculum.** Rollouts after teacher forcing, score-token dropout, and DAgger training. The piano-roll visualizations show the input performance, ground-truth score, and model predictions for Bach's *Fugue in C major*, BWV 846, from *The Well-Tempered Clavier, Book I*, performed by Rui Shi.

We isolate the contribution of pretraining by carrying an identically sized randomly initialized model through the same stage-2 recipe and checkpoint-selection rule. The pretrained model achieves more than twice the strict note F1 of random initialization on the test set and is ahead at every matched test checkpoint in strict, scaled, and IOI F1. Compared with task-specific systems (Beyer & Dai, 2024; Zeng et al., 2026), the adapted foundation model achieves comparable or higher note-level accuracy while producing outputs at the reference score's time scale substantially more often.

We also revisit the evaluation of note-level rhythm quantization. Whole-song note F1 is sensitive to global time scale and accumulated *drift*, where one lost beat displaces every later note. We introduce complementary metrics that separate time-scale discrepancies from local rhythmic structure, while treating the musical work as the unit of inference (Section 4).

Contributions:

- **Adaptation of a musical foundation model to rhythm quantization.** We adapt a pretrained symbolic-music foundation model to whole-song rhythm quantization without adding task-specific parameters. Using interleaved performance and score events, AMT-Q conditions each score prediction on bounded future performance context. On the ASAP test set, it achieves a strict note F1 of 9.33% and an inter-onset-interval F1 of 77.93%, exceeding the strongest task-specific systems.

- **A training curriculum for robust autoregressive rollouts.** We study the degradation caused by model-generated context during whole-song decoding and introduce a three-stage curriculum consisting of teacher forcing, heavy score-token dropout, and DAgger training on model-generated contexts. Ablations across initializations, decoding modes, and controls matched for training updates characterize the contribution of each stage (Ross et al., 2011).
- **Novel evaluation metrics that separate distinct failure modes.** In addition to strict note-level F1, we propose scaled note F1, inter-onset-interval (IOI) F1, and scaled IOI F1. Together, these metrics distinguish errors in absolute timing, global time scale, and local rhythmic structure, revealing differences between systems that are obscured by strict F1 alone.

2 Related Work

Performance-to-score transcription. Early rhythm quantizers explicitly modeled score structure and expressive timing (Desain & Honing, 1989; Cemgil et al., 2000; Raphael, 2002; Nakamura et al., 2017). Neural systems learn from paired data: PM2S predicts beats, quantized timings, and score attributes with a convolutional-recurrent model (Liu et al., 2022); Beyer & Dai (2024) use an encoder-decoder Transformer with score-token dropout; and Zeng et al. (2026) jointly learn quantization and performance rendering. The T5-based system of Wachter et al. (2025) receives ground-truth beat timings at inference. AMT-Q instead adapts a pretrained decoder-only music model, using interleaved performance events for future context without adding task-specific parameters or requiring beat annotations at inference.

Music foundation models. Symbolic-music Transformers learn musical structure from large corpora (Huang et al., 2019; Hsiao et al., 2021; Thickstun et al., 2024; Yuan et al., 2024; Qu et al., 2025), but their generative objectives do not directly provide a performance-to-score mapping. We use AMT's pretrained anticipation mechanism to condition score generation on performance notes. Our matched random-initialization comparison tests the contribution of musical pretraining to this adaptation.

Exposure bias. Scheduled sampling replaces reference context with predictions during training (Bengio et al., 2015), whereas score-token dropout removes context (Beyer & Dai, 2024). Our curriculum uses dropout before DAgger (Ross et al., 2011): whole-song rollouts collect model-generated contexts, paired with reference-score targets and accumulated for retraining. This exposes the model to prediction errors that dropout alone does not reproduce.

Evaluation metrics. Note-level F1 matches pitch and onset, optionally requiring offset agreement (McLeod & Steedman, 2018). MUSTER reports pitch, missing-note, extra-note, onset, and offset errors after alignment, with rhythm-correction operations for onset errors (Nakamura et al., 2018). Cogliati & Duan (2017) compare notation attributes using an edit-based measure. MV2H separates multi-pitch, voice, meter, note-value, and harmony scores (McLeod & Steedman, 2018); a later extension uses dynamic time warping for non-aligned scores (McLeod, 2019). Our evaluation targets note-level rhythm: strict F1 preserves timeline agreement, scaled F1 permits one global ratio, and IOI F1 measures interval agreement without preserving sequence order (Section 4). These complementary measures separate scale and drift effects without evaluating complete notation quality.

3 Method: AMT-Q

Task and sequence layout. A performance is a sequence of notes $\mathbf{p}_i = (o_i^P, d_i^P, \pi_i)$ ordered by onset, with onset o_i and duration d_i in 10 ms bins and MIDI pitch π_i. The model emits one event per performed note, either a score note $\hat{\mathbf{s}}_i = (\hat{o}_i^S, \hat{d}_i^S, \hat{\pi}_i)$ or $\varnothing$ ("no score note"), with onsets and durations expressed in ASAP's *annotated beats*, using 50 bins per beat regardless of the notated beat unit. For training, each performed note i is paired with an unused score note $a(i)$ of the same pitch near its beat-interpolated time, or $a(i) = \varnothing$. A window covers W consecutive performed notes beginning at note j. Its prefix contains L performance events, followed by $W - L$ interleaved score-performance pairs represented as AMT onset-duration-pitch triplets:

$$\mathbf{x} = \Big[\underbrace{\mathbf{p}_j, \varnothing,\ \ldots,\ \mathbf{p}_{j+L-1}, \varnothing}_{\text{prefix: } L \text{ performance events}},\ \underbrace{\mathbf{s}_j, \mathbf{p}_{j+L},\ \ldots,\ \mathbf{s}_{j+W-L-1}, \mathbf{p}_{j+W-1}}_{\text{alternating body: } W-L \text{ score slots}}\Big]. \tag{1}$$

The window lengths and alignment tolerance are specified in Appendix A.

Training objective. Let B be a micro-batch of T-token sequences $\mathbf{x}$. For clean windows, $\mathbf{x}$ provides targets after transposing both streams and scaling the performance tempo (Appendix A). With parameters θ, the next-token distribution p_θ uses corrupted inputs $\tilde{\mathbf{x}}$. Let $M(\mathbf{x})$ contain excluded target positions; it is empty except for generated score context in DAgger windows (Appendix A.1). The supervised prediction positions are

$$V_B = \{(\mathbf{x}, t) : \mathbf{x} \in B,\ 1 \le t < T,\ t+1 \notin M(\mathbf{x})\}.$$

We minimize token-level cross-entropy over these positions:

$$\mathcal{L}_{\mathrm{CE}}(\theta; B) = -\frac{1}{|V_B|} \sum_{(\mathbf{x},t)\in V_B} \log p_\theta\big(x_{t+1} \mid \tilde{\mathbf{x}}_{\le t}\big). \tag{2}$$

Motivated by the data-augmentation strategy of Beyer & Dai (2024), we corrupt the conditioning sequence with performance timing jitter and *score-token dropout*; neither affects the targets. Aggregated DAgger windows are not transposed, tempo-scaled, or jittered. For score-token dropout, let n be the number of real score triplets in the alternating body and ρ the dropout fraction. We uniformly sample $\mathrm{round}(\rho n)$ triplets without replacement and let D contain their onset, duration, and pitch token positions. Using token-embedding matrix $\mathbf{E}$ and positional embedding $\mathbf{P}_t$, we form the input representation $\mathbf{h}_t^{(0)}$ by zeroing the selected token embeddings:

$$\mathbf{h}_t^{(0)} = \big(1 - \mathbf{1}[t \in D]\big)\, \mathbf{E}[\tilde{x}_t] + \mathbf{P}_t. \tag{3}$$

Dropped tokens retain their targets in the loss, as in word dropout (Bowman et al., 2016; Beyer & Dai, 2024). An optional anchor of strength λ penalizes the mean squared deviation of the N trainable parameters from reference weights θ^0. Each step minimizes $\mathcal{L}_{\mathrm{CE}} + \Omega$, with anchor penalty

$$\Omega(\theta) = \frac{\lambda}{N} \sum_{i=1}^{N} \big(\theta_i - \theta_i^0\big)^2. \tag{4}$$

3.1 Curriculum

Our three-stage curriculum progresses from clean to corrupted and model-generated score contexts (Figure 1). All optimization, augmentation, and sampling settings are given in Appendix A and Table 3.

Stage 1: Teacher forcing. We fine-tune all pretrained AMT parameters on matched-pair interleaved performance-score windows using the next-token objective in Eq. 2. The checkpoint with the lowest validation loss initializes Stage 2.

Stage 2: Score-token dropout. We continue training with score-token dropout (Eq. 3), yielding AMT-Q. We select the checkpoint by whole-song validation strict F1, using the test-time decoder (Section 3.2) with dropout disabled. These rollouts use unfiltered performances, so selection reflects the autoregressive setting rather than performance on clean training windows.

Stage 3: DAgger. Dropout removes context but does not reproduce incorrect predictions. Following DAgger (Ross et al., 2011), we collect contexts generated by the current model and train it to continue from them using reference-score targets. Collection uses unfiltered training performances, including notes without an aligned score partner.

At each sliding-window boundary, we retain the model's recent score predictions as context and construct targets for the remaining score slots. To express these targets in the model's current clock, we subtract the median reference-to-prediction onset offset among matched context notes. Shifted

reference onsets are clipped at zero; reference durations and pitches are unchanged, and unmatched performed notes receive $\varnothing$ targets. Generated context triplets are masked from the loss, so they serve as inputs rather than prediction targets. Appendix A.1 gives the exact target construction.

Each round adds its harvested windows to all previous harvests and a fixed set of clean matched-pair windows. Rollout collection uses the preceding round's model, but retraining always restarts from the same Stage 2 checkpoint. Score-token dropout applies to harvested windows and is disabled on clean windows. The final run is selected by whole-song validation strict F1.

3.2 Inference: sliding-window decoding

We decode unfiltered performance MIDI in overlapping windows to accommodate AMT's finite context. Within each window, we greedily generate score triplets while restricting each argmax to the permitted vocabulary range for the corresponding score-token type. After each score triplet is generated, the next performance triplet is appended, maintaining the alternating structure in Eq. 1 and advancing the lookahead.

The first window generates score events until the context is full. We then re-pack the window with a fresh performance prefix and the most recent generated score triplets, leaving room to continue decoding. Window sizes and the retained-context length are given in Appendix A.

We subtract the earliest retained predicted onset from score onsets, leaving durations unchanged; performance onsets are independently re-centered at zero. New predictions return to the whole-song timeline by adding back the retained anchor's global onset. Only newly generated notes are appended to the output.

Figure 6 in Appendix E compares engravings of the input performance, quantized output, and reference score.

4 Evaluating Note-level Rhythm Quantization

Data, baselines, and inference. We evaluate whole-song rhythm quantization using exactly the training, validation, and test split of Beyer & Dai (2024) and Zeng et al. (2026). The split contains 822 training, 86 validation, and 59 test performances, covering 176, 16, and 14 distinct musical works, respectively. Both baselines are run from released weights; their quarter-note output is mapped onto the reference grid using the score's own quarter-to-beat map (Appendix C). PM2S (Liu et al., 2022) is not compared because its released model was trained on a split of ASAP containing 55 of our 59 test performances.

Because multiple performances come from the same musical work, we treat the work as the unit of inference. p_{work} is computed using an exact sign-flip test on the per-work mean differences (14 works on test; 16 on validation), whereas p_{perf} uses a sign-flip test on individual performance-level differences. "Over works" denotes the unweighted mean of the per-work means.

Metrics. *Strict F1* matches predicted notes to reference notes on the whole-song timeline: pitch must agree, onset must lie within ± 1 bin, and matching is one-to-one; each note list is first shifted to begin at time zero.

Scaled F1 maximizes strict F1 over a fixed set $\mathcal{R}$ of 13 global ratios between the beat units annotated by ASAP, ranging from $\frac{1}{6}$ to 6:

$$F_{\text{scale}}(\hat{S}, S) = \max_{r \in \mathcal{R}} F^{(r)}_{\text{note}}(\hat{S}, S).$$

Figure 2 illustrates the effect of correcting a global time-scale mismatch.

IOI F1 replaces each onset by the interval between the current note and the previous note and matches (interval, pitch) pairs in an order-free manner, making the metric insensitive to accumulated global drift.

Scaled IOI F1 maximizes this quantity over the same ratio set:

$$F_{\text{IOI,scale}}(\hat{S}, S) = \max_{r \in \mathcal{R}} F^{(r)}_{\text{IOI}}(\hat{S}, S).$$

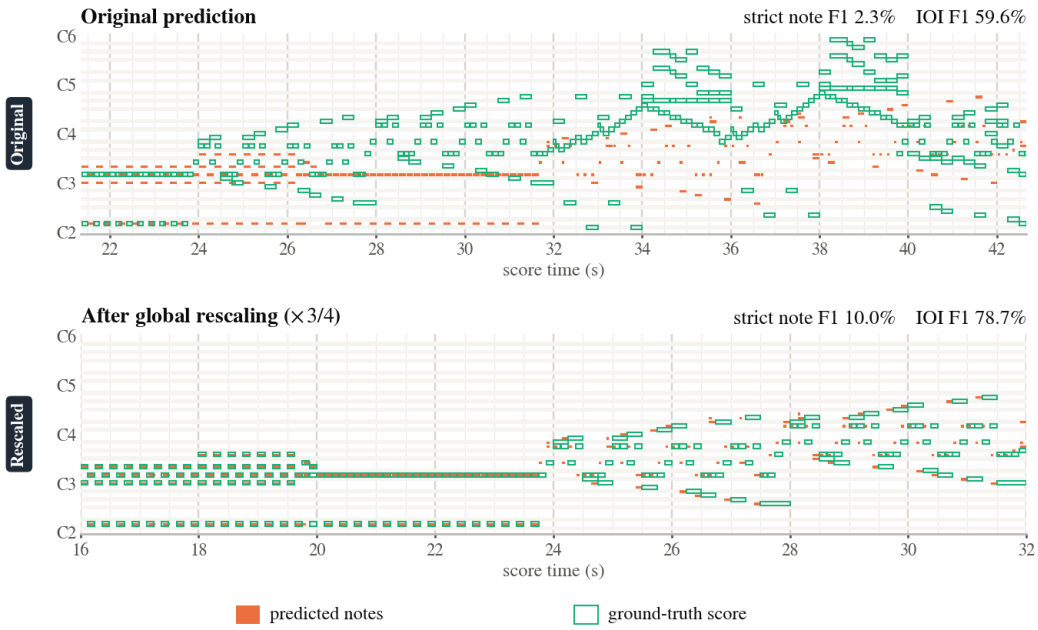


Figure 2: **Example of a global time-scale mismatch.** Predictions for Prokofiev's *Toccata*, performed by Julia Kociuban, shown before (top) and after (bottom) rescaling all predicted onsets and durations by $\frac{3}{4}$.

The *local time scale* of an output is the Theil–Sen slope (Sen, 1968) of its predicted onsets against aligned reference onsets over note pairs separated by 1 to 4 beats. We classify an output as *at the reference scale* when this slope is closer in log space to 1 than to any other ratio in $\mathcal{R}$, corresponding to the interval 0.87–1.15. Rescaling an output once by the ratio nearest in log space to the reciprocal of its measured scale gives strict and IOI F1 *at the estimated scale*, without maximizing over ratios. Detailed metric definitions and reference quantizers appear in Appendix C.

Sensitivity of whole-song F1 to scale and drift. The controlled perturbations in Table 6 (Appendix C) show that strict F1 is highly sensitive to global scaling and accumulated timing offsets. A single lost beat reduces strict F1 by 39 points, while a factor-two global scale change reduces it by 96 points. By comparison, local note-value errors and missing notes reduce F1 by roughly the fraction of notes that were perturbed.

The same distinction appears on model outputs (Table 1). An oracle that knows only each performance's average tempo and rounds to a quarter-beat grid reaches 9.1% strict F1, comparable to the learned systems (7.6–9.3%); over works it is significantly above Zeng et al. (2026) ($p_{\text{work}} = 0.028$). At the same time, AMT-Q and both learned baselines exceed this oracle on scaled IOI F1 over works ($p_{\text{work}} \leq 0.028$). Strict whole-song F1 therefore combines several distinct sources of error that the complementary metrics separate.

5 Results

5.1 Pretraining enables transfer learning

Pretraining comparison. The random-initialization control uses the identical Stage 2 recipe and checkpoint-selection rule as AMT-Q: the checkpoint with the highest whole-song validation strict F1 (Appendix B). On test, the pretrained model is ahead on every reported metric (Table 4, Appendix B). It more than doubles strict F1, from 3.23% to 8.72%, and has higher strict F1 on 11 of the 14 works ($p_{\text{work}} = 0.004$). Its IOI F1 is 6.4 points higher per performance and 7.0 points higher over works ($p_{\text{work}} = 0.033$).

Table 1: **Whole-song results on the 59 test performances of 14 works** (%). Strict F1 is reported as a mean over performances (median) and as the mean of the 14 per-work means with a 95% bootstrap interval over works; the remaining F1 metrics are means over performances. Global shift is strict F1 after the best single whole-song onset shift. Ref. scale counts outputs whose measured local time scale matches the reference scale. Reference quantizers use ground-truth tempo or beat information. AMT-Q and the random-initialization control use the same validation-selection rule; AMT-Q + DAgger is the validation-selected DAgger run (Appendix A). Bold indicates the largest point estimate among learned models in each column.

	Strict F1		Scaled	IOI	Scaled	Global	Ref.
System	mean (median)	over works [95% CI]	F1	F1	IOI F1	shift	scale
Reference quantizers (no learning)							
Performance clock	1.13 (0.88)	1.1 [0.8, 1.3]	2.73	50.32	62.03	2.50	9
Oracle global tempo, $\frac{1}{4}$ grid	9.14 (8.05)	10.2 [7.5, 13.2]	9.39	62.16	77.28	13.65	48
Oracle beats, $\frac{1}{12}$ grid	61.61 (58.72)	64.4 [57.2, 71.4]	61.61	72.50	73.27	63.00	59
Task-specific systems, trained from scratch							
Beyer & Dai (2024)	8.96 (5.47)	9.2 [6.1, 12.6]	**11.54**	67.81	**83.02**	16.96	23
Zeng et al. (2026)	7.58 (4.56)	7.2 [4.8, 10.3]	10.79	63.04	80.29	12.82	16
Adapted AMT architecture							
Teacher forcing only (stage 1)	3.94 (3.06)	5.2 [3.9, 6.7]	6.85	61.44	78.11	7.65	6
Random init., same stage 2	3.23 (3.03)	3.7 [2.7, 5.0]	6.36	72.43	80.24	15.93	36
AMT-Q (pretrained, stage 2)	8.72 (5.14)	9.6 [5.8, 14.8]	10.79	**78.81**	82.24	20.77	**43**
AMT-Q + DAgger	**9.33** (4.75)	**14.7** [5.8, 28.5]	11.20	77.93	81.65	**21.96**	41

Checkpoint robustness. The pretraining advantage is not specific to the selected checkpoint. We decoded both runs on test at regular training intervals (Figure 3(b); schedule in Appendix B). The pretrained run is ahead at all ten matched checkpoints in strict, scaled, and IOI F1, and at nine of ten in scaled IOI F1, trailing by 0.1 points at the tenth. Averaged across these checkpoints, it leads by 2.9 strict-F1 points and 5.6 IOI-F1 points, with higher values on 12 and 12 of the 14 works, respectively ($p_{\text{work}} = 0.004$ and 0.002).

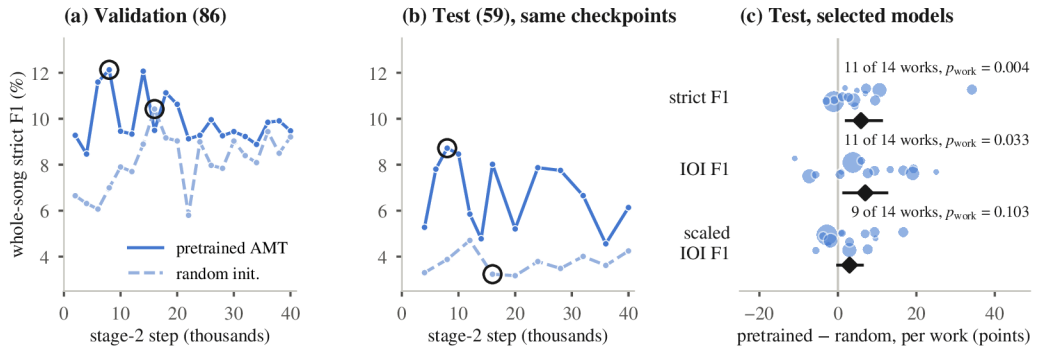


Figure 3: **Pretraining versus random initialization.** Pretrained AMT (solid) and the same architecture initialized at random (dashed), fine-tuned using the identical Stage 2 recipe. (a) Whole-song strict note F1 for all evaluated checkpoints on the 86 validation performances; the selected checkpoint from each run is ringed. (b) The corresponding checkpoints evaluated on the 59 test performances. (c) Test-set differences between the two selected models by work, with dot area proportional to the number of performances and the mean over the 14 works with its 95% bootstrap interval shown as a diamond.

On validation, where the checkpoints were selected, the difference between the selected models is smaller: $+1.7$ strict-F1 points and not significant over works. The pretrained run nevertheless has higher strict F1 at 19 of 20 matched checkpoints (Figure 3(a)) and is ahead at 13–17 checkpoints on

the other metrics. Its validation IOI and scaled-IOI gains are significant per performance ($p_{\text{perf}} \leq 0.013$) but not over works ($p_{\text{work}} = 0.14$ and 0.13; Table 5).

Music versus text pretraining. We additionally initialize the same pipeline from SmolLM2, a 362M-parameter pretrained text model (initialization details in Appendix A). Under the AMT training recipe, SmolLM2 reaches 3.9% strict F1 (Table 9, arm B6) and trails AMT-Q by 14 IOI-F1 points over works (Figure 5). Thus, initialization from a generic pretrained Transformer does not by itself account for the gain from AMT pretraining. Each initialization arm consists of one training run, and the random and pretrained models enter Stage 2 from different Stage 1 runs (Appendix B).

5.2 Performance against task-specific systems

Time-scale behavior. AMT-Q's measured local time scale matches the reference on 43 of 59 test outputs, compared with 23 for Beyer & Dai and 16 for Zeng et al. (Figure 4(a); per-performance $p = 0.002$ and $p < 0.001$; over works, $p_{\text{work}} = 0.34$ and 0.043). This difference accounts for most of AMT-Q's advantage in IOI F1. On the 36 performances for which Beyer & Dai writes at a different time scale, AMT-Q has higher IOI F1 on 34 and leads by 20 points on average. On performances for which Beyer & Dai is at the reference scale, AMT-Q does not have an IOI advantage (Figure 4(b)).

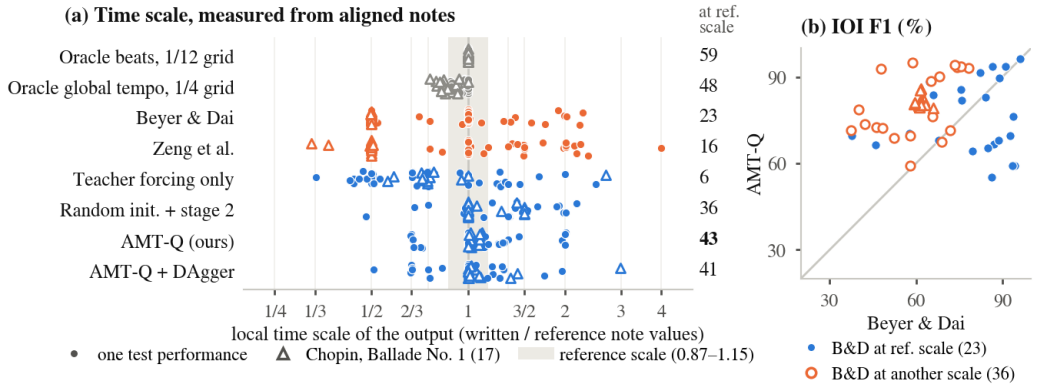


Figure 4: **Local time-scale estimates and IOI F1.** (a) Local time scale of every test output: the slope of predicted against aligned reference onsets over notes separated by 1–4 beats (log axis; shaded band: reference scale; right: count within the band). Beyer & Dai writes every performance of *Ballade No. 1* (triangles) at approximately half the reference note values and Zeng et al. does so for 15 of 17, whereas AMT-Q writes these performances at the reference scale. (b) IOI F1 of AMT-Q against Beyer & Dai by performance. Open points denote performances on which Beyer & Dai is not at the reference scale; filled points denote those on which it is.

After correcting each output for its measured time scale, only 1.5 points of the 11.0-point per-performance IOI difference between AMT-Q and Beyer & Dai remain; under scaled IOI F1, the systems are similarly close (Table 4). Most of the raw IOI advantage therefore reflects time-scale agreement. Predicting in annotated beats alone does not guarantee this agreement: the same architecture after teacher forcing produces only 6 outputs at the reference scale; a Stage 1 variant without the anchor and selected on rollouts produces 22.

Note-level F1. On strict F1, AMT-Q is not statistically distinguishable from Beyer & Dai or Zeng et al. under the tests reported here (Table 4). The 90% interval for its difference from Beyer & Dai lies within ± 2.4 points per performance and ± 4.2 points over works. Thus, a model pretrained for symbolic-music generation reaches the note-level accuracy of architectures designed specifically for rhythm quantization.

This result concerns the selected checkpoint. Across the 13 Stage 2 checkpoints decoded on test, strict F1 ranges from 4.6% to 8.7% (Figure 3(b)), which motivates the all-checkpoint comparison used for the pretraining analysis in Section 5.1. Medians and pairwise win counts favor Beyer &

Dai (37 of 59 performances), and permitting one global onset shift per song, which can forgive an early offset but not drift, also does not separate the systems.

MUSTER evaluation. We evaluate a common subset of 52 performances from 13 works, excluding all seven *Ondine* performances because MUSTER's matcher crashes for every system, including the reference. All predictions and the reference pass through one shared MusicXML writer; export corrections and an offset-interpolation patch are detailed in Appendix C.1.

Under this protocol, AMT-Q's mean MUSTER error is statistically indistinguishable from Beyer & Dai over works ($p_{\text{work}} = 0.27$) and lower than Zeng et al.'s ($p_{\text{work}} = 0.003$). Table 2 reports all systems and a reference self-comparison alongside the note-level evaluation.

Table 2: **MUSTER evaluation on the common scorable subset** (52 performances, 13 works). Lower is better. Bold indicates the best result and italics the second-best result among the four evaluated systems.

Method	E_{p}	E_{miss}	E_{extra}	E_{onset}	E_{offset}	E_{avg}
Beyer & Dai (2024)	2.43	6.49	**3.52**	33.31	56.33	20.42
Zeng et al. (2026)	2.61	8.03	*4.50*	37.52	60.23	22.58
Ours, Stage 2	**1.93**	**4.42**	4.89	*31.10*	*55.65*	**19.60**
Ours + DAgger	*1.95*	*5.60*	6.03	**30.80**	**55.43**	*19.96*
Reference self-comparison	0.00	0.00	1.84	0.00	1.91	0.75

5.3 Ablation study

A complete account of the ablations, including Figure 5, is given in Appendix D. We focus on corrupted-context training and checkpoint selection.

Corrupted-context training. Teacher forcing alone produces 6 of 59 outputs at the reference time scale (Table 1) and trails AMT-Q by 19 IOI-F1 points over works ($p_{\text{work}} = 0.001$; Figure 5). Removing score-token dropout or replacing Muon with AdamW reduces IOI F1 by approximately 6 points over works, with much of the difference associated with time scale. Both ablation arms used shorter training budgets than AMT-Q (Appendix D).

Checkpoint selection. The final Stage 2 checkpoint, rather than the checkpoint selected by whole-song validation strict F1, scores 2.6 strict-F1 points below AMT-Q per performance on test ($p_{\text{perf}} = 0.008$; $p_{\text{work}} = 0.25$). A DAgger round initialized from this final checkpoint trails one initialized from AMT-Q by 11 strict-F1 points and 15 IOI-F1 points over works ($p_{\text{work}} \leq 0.022$). Although we select checkpoints using whole-song validation rollouts, validation and test rank the decoded checkpoints almost independently (Spearman 0.02).

6 Conclusion

We present a method for adapting a pretrained music foundation model to rhythm quantization without adding task-specific parameters. The model represents performances and scores as an interleaved sequence. Our three-stage curriculum overcomes exposure bias by progressing from teacher-forced contexts to heavily corrupted contexts and finally to contexts generated by the model itself. Our experiments show that symbolic-music pretraining substantially improves this adaptation relative to random initialization and generic text-model pretraining.

We also introduce scaled and inter-onset-interval variants of note F1 to distinguish global time-scale discrepancies, accumulated drift, and local rhythmic agreement. On the shared ASAP test set, the adapted model reaches the note-level accuracy of task-specific systems while producing outputs at the reference time scale substantially more often. More broadly, our results illustrate the potential for music foundation models in the data-limited regime as transferable priors.

AI USE STATEMENT

We used large language model assistants as coding assistants for implementing the method and for writing analysis and plotting scripts. We also used them to draft and edit the paper and to review the clarity of the methodological descriptions and the interpretation of reported results. They were not used to generate research ideas, experimental results, or datasets. Reported numerical results were computed from logged experiment outputs using analysis scripts. The authors take responsibility for all content.

ETHICS STATEMENT

This work uses the public ASAP dataset and a model pretrained on the public Lakh MIDI dataset. It involves no human-subject experiments or new data collection. The underlying test compositions are in the public domain.

REPRODUCIBILITY STATEMENT

Section 3 and Appendix A specify the model, sequence layout, objectives, training settings, and decoder. Appendix A.1, including Algorithm 1, details DAgger data collection and retraining. We use the public ASAP split of Beyer & Dai (2024); Appendix B provides checkpoint-selection details. Section 4 and Appendix C describe the metrics, reference quantizers, time-scale estimates, controlled perturbations, and statistical tests. Appendix C.1 documents the shared notation writer and MUSTER corrections. Upon acceptance, we will publicly release the code, model checkpoints, per-performance evaluation outputs, scoring and statistical-analysis scripts with their seeds, and the MUSTER patch.

REFERENCES

Loubna Ben Allal, Anton Lozhkov, Elie Bakouch, Gabriel Martín Blázquez, Guilherme Penedo, Lewis Tunstall, Andrés Marafioti, Hynek Kydlíček, Agustín Picón Lajarín, Vaibhav Srivastav, et al. SmolLM2: When smol goes big – data-centric training of a small language model. *arXiv preprint arXiv:2502.02737*, 2025.

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks. In *Advances in Neural Information Processing Systems*, volume 28, pp. 1171–1179, 2015.

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, San Francisco, CA, USA, 2024. doi: 10.5281/zenodo.14877339.

Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew M. Dai, Rafal Jozefowicz, and Samy Bengio. Generating sentences from a continuous space. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning (CoNLL)*, pp. 10–21, 2016. doi: 10.18653/v1/K16-1002.

Ali Taylan Cemgil, Peter Desain, and Bert Kappen. Rhythm quantization for transcription. *Computer Music Journal*, 24(2):60–76, 2000. doi: 10.1162/014892600559218.

Andrea Cogliati and Zhiyao Duan. A metric for music notation transcription accuracy. In *Proceedings of the 18th International Society for Music Information Retrieval Conference (ISMIR)*, 2017. doi: 10.5281/zenodo.1415830.

Peter Desain and Henkjan Honing. The quantization of musical time: A connectionist approach. *Computer Music Journal*, 13(3):56–66, 1989. doi: 10.2307/3680012.

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.

Wen-Yi Hsiao, Jen-Yu Liu, Yin-Cheng Yeh, and Yi-Hsuan Yang. Compound word transformer: Learning to compose full-song music over dynamic directed hypergraphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 178–186, 2021. doi: 10.1609/aaai.v35i1.16091.

Cheng-Zhi Anna Huang, Ashish Vaswani, Jakob Uszkoreit, Ian Simon, Curtis Hawthorne, Noam Shazeer, Andrew M. Dai, Matthew D. Hoffman, Monica Dinculescu, and Douglas Eck. Music transformer: Generating music with long-term structure. In *International Conference on Learning Representations (ICLR)*, 2019. URL https://openreview.net/forum?id=rJe4ShAcF7.

Keller Jordan, Yuchen Jin, Vlado Boza, Jiacheng You, Franz Cesista, Laker Newhouse, and Jeremy Bernstein. Muon: An optimizer for hidden layers in neural networks. Blog post, 2024. URL https://kellerjordan.github.io/posts/muon/.

Konwoo Kim, Suhas Kotha, Percy Liang, and Tatsunori Hashimoto. Pre-training under infinite compute. In *International Conference on Learning Representations (ICLR)*, 2026. URL https://openreview.net/forum?id=ck0aZTAnwK.

Roman Koshkin, Katsuhito Sudoh, and Satoshi Nakamura. TransLLaMa: LLM-based simultaneous translation system. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, 2024.

Jingyuan Liu, Jianlin Su, Xingcheng Yao, Zhejun Jiang, Guokun Lai, Yulun Du, Yidao Qin, Weixin Xu, Enzhe Lu, Junjie Yan, Yanru Chen, Huabin Zheng, Yibo Liu, Shaowei Liu, Bohong Yin, Weiran He, Han Zhu, Yuzhi Wang, Jianzhou Wang, Mengnan Dong, Zheng Zhang, Yongsheng Kang, Hao Zhang, Xinran Xu, Yutao Zhang, Yuxin Wu, Xinyu Zhou, and Zhilin Yang. Muon is scalable for LLM training. *arXiv preprint arXiv:2502.16982*, 2025. URL https://arxiv.org/abs/2502.16982.

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

Mingbo Ma, Liang Huang, Hao Xiong, Renjie Zheng, Kaibo Liu, Baigong Zheng, Chuanqiang Zhang, Zhongjun He, Hairong Liu, Xing Li, Hua Wu, and Haifeng Wang. STACL: Simultaneous translation with implicit anticipation and controllable latency using prefix-to-prefix framework. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 3025–3036, 2019.

Andrew McLeod. Evaluating non-aligned musical score transcriptions with MV2H. *arXiv preprint arXiv:1906.00566*, 2019. doi: 10.48550/arXiv.1906.00566.

Andrew McLeod and Mark Steedman. Evaluating automatic polyphonic music transcription. In *Proceedings of the 19th International Society for Music Information Retrieval Conference (ISMIR)*, pp. 42–49, 2018.

Eita Nakamura and Kazuyoshi Yoshii. Musical rhythm transcription based on Bayesian piece-specific score models capturing repetitions. *Information Sciences*, 572:482–500, 2021. doi: 10.1016/j.ins.2021.04.100.

Eita Nakamura, Kazuyoshi Yoshii, and Shigeki Sagayama. Rhythm transcription of polyphonic piano music based on merged-output HMM for multiple voices. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 25(4):794–806, 2017. doi: 10.1109/TASLP.2017.2662479.

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)*, pp. 101–105. IEEE, 2018. doi: 10.1109/ICASSP.2018.8461914.

Xingwei Qu, Yuelin Bai, Yinghao Ma, Ziya Zhou, Ka Man Lo, Jiaheng Liu, Ruibin Yuan, Lejun Min, Xueling Liu, Tianyu Zhang, Xeron Du, Shuyue Guo, Yiming Liang, Yizhi Li, Shangda Wu, Junting Zhou, Tianyu Zheng, Ziyang Ma, Fengze Han, Wei Xue, Gus Xia, Emmanouil Benetos, Xiang Yue, Chenghua Lin, Xu Tan, Wenhao Huang, Jie Fu, and Ge Zhang. MuPT: A generative symbolic music pretrained transformer. In *International Conference on Learning Representations (ICLR)*, 2025. URL https://openreview.net/forum?id=iAK9oHp4Zz.

Christopher Raphael. A hybrid graphical model for rhythmic parsing. *Artificial Intelligence*, 137 (1–2):217–238, 2002. doi: 10.1016/S0004-3702(02)00192-3.

Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 15 of *Proceedings of Machine Learning Research*, pp. 627–635. PMLR, 2011.

Pranab Kumar Sen. Estimates of the regression coefficient based on kendall's tau. *Journal of the American Statistical Association*, 63(324):1379–1389, 1968. doi: 10.1080/01621459.1968.10480934.

John Thickstun, David Hall, Chris Donahue, and Percy Liang. Anticipatory music transformer. *Transactions on Machine Learning Research*, 2024. URL https://openreview.net/forum?id=EBNJ33Fcrl.

Maximilian Wachter, Sebastian Murgul, and Michael Heizmann. Transformer-based rhythm quantization of performance MIDI using beat annotations. In *2025 International Conference on Smart Multimedia (ICSM)*, pp. 1–7. IEEE, 2025. doi: 10.1109/ICSM64417.2025.11541508.

Ruibin Yuan, Hanfeng Lin, Yi Wang, Zeyue Tian, Shangda Wu, Tianhao Shen, Ge Zhang, Yuhang Wu, Cong Liu, Ziya Zhou, Liumeng Xue, Ziyang Ma, Qin Liu, Tianyu Zheng, Yizhi Li, Yinghao Ma, Yiming Liang, Xiaowei Chi, Ruibo Liu, Zili Wang, Chenghua Lin, Qifeng Liu, Tao Jiang, Wenhao Huang, Wenhu Chen, Jie Fu, Emmanouil Benetos, Gus Xia, Roger Dannenberg, Wei Xue, Shiyin Kang, and Yike Guo. ChatMusician: Understanding and generating music intrinsically with LLM. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 6252–6271, Bangkok, Thailand, 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.373.

Mingliang Zeng, Xu Tan, Rui Wang, Zeqian Ju, Tao Qin, and Tie-Yan Liu. MusicBERT: Symbolic music understanding with large-scale pre-training. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pp. 791–800, 2021.

Wei Zeng, Junchuan Zhao, and Ye Wang. Bridging piano transcription and rendering via disentangled score content and style. In *International Conference on Learning Representations (ICLR)*, 2026. URL https://openreview.net/forum?id=173Pq3F31r.

A IMPLEMENTATION DETAILS

Table 3 summarizes the optimization and training settings for the curriculum in Section 3.1. The remaining representation, augmentation, decoding, and rollout-collection settings are specified below.

Table 3: **Hyperparameters.** All stages fine-tune all 360M parameters of AMT `music-medium-800k`, a GPT-2-style decoder with a 1,024-token context and a 55,028-token vocabulary. The effective batch size is 64 windows of 1,020 tokens throughout, and the loss is Eq. 2, plus Eq. 4 when the anchor is enabled.

	Stage 1	Stage 2 (AMT-Q)	Stage 3 (each round)
Initialization	pretrained AMT	stage 1, step 2,500	AMT-Q (stage 2, step 8,000)
Data	clean windows	clean windows	36,000 clean windows + all harvested windows
Augmentation	transposition, tempo and timing jitter (below)	same	clean windows as in stage 1; harvested windows none
Optimizer	AdamW, lr 3×10^{-5}, wd 0.01, clip 2.0	Muon (hidden matrices), lr 2×10^{-4}, wd 0.5; Adam lr 3×10^{-5} for embeddings (wd 0.05) and 1-D parameters	AdamW, lr 3×10^{-5}
Schedule	cosine to 10%, no warm-up	cosine to 10%	cosine to 10%
Steps	20,000	40,000	5,000
Anchor λ, θ^0	10^5, pretrained weights	0 (off)	10^5, AMT-Q's weights
Dropout ρ	0	0.75	0.75 harvested / 0 clean
Loss mask $M(\mathbf{x})$	none	none	69 retained score slots of harvested windows
Selection	lowest validation loss (step 2,500 of 20,000)	whole-song validation strict F1 (20 checkpoints)	fixed 5k steps; run chosen by validation strict F1 among 3 runs

Window layout. In Eq. 1, $W = 170$ performed notes and $L = 32$ prefix performance events give 138 score slots. The prefix pairs each performance event with a $\varnothing$ placeholder; the alternating body pairs each score event with a performance event 32 positions later. Each score prediction therefore sees its corresponding performed note and the following 31 performed notes. Each window contains 1,020 tokens within AMT's 1,024-token context.

Optimization. Stage 1 uses the AdamW settings in Table 3, with a warmup-free cosine schedule from 3×10^{-5} to 3×10^{-6} and the anchor in Eq. 4 toward the pretrained weights. Global gradient-norm clipping uses a threshold of 2.0. Stage 2 uses Muon (Jordan et al., 2024) for hidden weight matrices because of its reported sample efficiency (Liu et al., 2025); embeddings and one-dimensional parameters use Adam. The large decoupled weight decay follows Kim et al. (2026). Stage 2 disables the anchor, and score-token dropout is disabled during validation.

Data and loss. Onsets and durations are quantized to 10 ms bins, and score time is normalized to 0.5 s per annotated beat, so one score bin is $1/50$ of an annotated beat regardless of the notated beat unit. For example, a sixteenth note under a quarter-note beat corresponds to 12.5 bins; the reference therefore rounds such an onset by at most half a bin, within the ±1-bin evaluation tolerance.

Clean windows slide one note at a time over each training performance after unmatched performed notes have been removed, yielding 2,154,473 windows. Transposition of up to ±12 semitones applies to both streams; tempo scaling of up to $\pm20\%$, onset jitter (sd 0.05), and duration jitter (0.05) apply to the performance only. Transposition and tempo scaling are applied to both inputs and targets, whereas jitter is applied only to the input. The loss in Eq. 2 is the token-weighted mean over all supervised targets in a micro-batch, averaged over gradient-accumulation micro-batches and data-parallel ranks; Stage 1 validation loss is this all-token loss.

Score-token dropout never selects prefix placeholders or ∅ targets, and position embeddings are still added at dropped positions. An early run combining a larger learning rate and decay with the anchor degraded because the anchor opposed the decay, so Stage 2 is run without the anchor.

Alignment and unmatched notes. The alignment pairs each performed note with an unused score note of the same pitch within 0.1 s of its beat-interpolated time, using ASAP's beat annotations. It is used to build the clean windows, construct DAgger targets, and estimate time scale; strict F1 does not use the alignment.

On the 59 test performances, 15.1% of performed notes (33,305 of 220,287) have no partner, ranging by work from 2.3% for Beethoven to 25.6% for *Ondine*. On the 150 training performances harvested for DAgger, the unmatched rate is 8.2%. Stages 1–2 never see these unmatched notes but must nevertheless emit an output for each input note at test time. Across test performances, the unmatched rate correlates with MUSTER's extra-note rate for AMT-Q (Spearman $+0.38$, $p = 0.005$, $n = 52$) and for Beyer & Dai ($+0.29$, $p = 0.039$), but not with AMT-Q's excess extra-note rate over Beyer & Dai ($+0.11$, $p = 0.42$). We have not separately ablated this train-test mismatch.

Decoding and cost. We use the sliding-window procedure in Section 3.2. Decoding is greedy. Each generated token is restricted to the vocabulary range corresponding to its slot role (onset, duration, or pitch), and the pitch range includes ∅, which is removed before scoring. The first window generates 138 score events; each re-pack retains 69 generated events as context and produces 69 new events.

For each 69 performed notes, the decoder re-primes 606 tokens (the 32-pair prefix and 69 retained pairs), then decodes 207 score tokens one at a time while feeding 207 control tokens through the cache. This corresponds to approximately 9 prefill tokens, 3 autoregressive decoding steps, and one 3-token control feed per performed note. We did not benchmark wall-clock time against the baselines. The *reset* decoding arm (C1 in Table 9) restarts each window from a bare 32-note prefix and re-anchors its onsets at the largest onset predicted so far.

Initializations. *Random initialization* uses the AMT architecture with freshly initialized weights. Its Stage 2 arm (B8) uses the identical Stage 2 recipe beginning from the random-initialization Stage 1 checkpoint at step 20,000, with the anchor disabled. Like AMT-Q, it retains the best of 20 checkpoints according to whole-song validation strict F1, selecting step 16,000.

SmolLM2-360M (Ben Allal et al., 2025) is a Llama-style text decoder with 32 layers, width 960, rotary positional embeddings, and 362M parameters. Its embedding matrix is resized to our vocabulary. In the full-recipe arm B6, the tied embedding is re-initialized while the pretrained Transformer body is retained. We use the AMT training recipe rather than retuning it for SmolLM2; the Stage 2 checkpoint at step 36,000 is selected from 20 checkpoints using validation rollouts. The Stage 1-only arm A3 retains the text embedding rows and the 10^5 anchor, whereas A1 and A2 run without the anchor. Each Stage 1-only arm selects among checkpoints at 2.5k, 5k, 10k, and 20k steps using whole-song validation F1; all select 20k.

A.1 DAgger data collection and targets

This section gives the target construction and round schedule for Stage 3 in Section 3.1; Algorithm 1 summarizes one complete training run.

Let π_0 = AMT-Q and let π_r denote the model after round r. In each round, π_{r-1} greedily decodes the same 150 unfiltered training performances using the whole-song inference procedure in Section 3.2. At each re-pack boundary b, we save a window with the model's 69 score predictions retained from the preceding window as conditioning context.

We retain only boundaries with a full 32-note performance prefix. To construct targets for the next 69 score slots, we first align the reference score to the model's current clock. Let O_b index retained score slots with a non-null prediction and an aligned reference note. We skip boundaries with $O_b = \varnothing$. Otherwise, using predicted onsets in the re-centered window clock, we estimate the offset by

$$\delta_b = \mathrm{median}_{k \in O_b} \left(o^S_{a(k)} - \hat{o}_k \right).$$

We then translate each reference onset by this offset. For the next 69 positions,

$$y_k^* = \begin{cases} \left(\max\left(0, o^S_{a(k)} - \delta_b\right), d^S_{a(k)}, \pi_{a(k)}\right), & a(k) \neq \varnothing, \\ \varnothing, & \text{otherwise}, \end{cases} \qquad k = b+69, \ldots, b+137. \tag{5}$$

When the context window becomes full, we retain the 69 most recent generated score triplets together with the corresponding performance context and use the reference score to construct targets for the remaining 69 score slots. The loss is masked over the generated triplets, which serve only as conditioning inputs. Let $\mathcal{H}_r$ denote the windows harvested in round r, and let $\mathcal{D}_{\text{clean}}$ denote the fixed corpus of clean teacher-forced windows constructed from matched pairs. The round-r training set is

$$\mathcal{D}_r = \mathcal{D}_{\text{clean}} \cup \bigcup_{j=1}^{r} \mathcal{H}_j.$$

The first round uses the Stage 2 model to collect rollouts, while subsequent rounds use the model from the preceding round. Each retraining round nevertheless restarts from the same Stage 2 checkpoint, holding initialization fixed while the collected dataset grows.

We train for $5{,}000$ steps per round and perform two rounds. The fraction of training examples containing model-generated context grows from approximately one in seven in round 1 to one in four in round 2. Retraining uses AdamW at 3×10^{-5} with an L_2 penalty anchored to the Stage 2 weights. Score-token dropout remains at 75% on harvested windows and is disabled on clean matched-pair windows. Stage 3 therefore mixes clean reference contexts with generated contexts from unfiltered performances.

The clean corpus contains 36,000 matched-pair windows. Runs a, b, and c are three training replicates: a uses four GPUs without a fixed seed, while b and c use two GPUs with seeds 1 and 2. They share the same deterministic first harvest; later harvests use each run's preceding model. We select among their round-2 models by whole-song validation strict F1, using the fixed budget in Table 3. Third-round results in Appendix D are supplementary ablations.

Algorithm 1 DAgger training for one run

Inputs: Stage 2 model π_0; fixed clean corpus $\mathcal{D}_{\text{clean}}$; the same 150 unfiltered training performances and aligned reference scores.

$\mathcal{D} \leftarrow \mathcal{D}_{\text{clean}}$ (36,000 windows)
for $r = 1, 2$ **do**
  $\mathcal{H}_r \leftarrow \varnothing$
  for each training performance **do**
    Greedily decode with π_{r-1} using Section 3.2.
    for each re-pack boundary b with a full 32-note prefix **do**
      Save the re-packed window context, including 69 retained score predictions.
      $O_b \leftarrow$ retained positions with a non-null prediction and aligned reference.
      if $O_b = \varnothing$ **then continue**.
      $\delta_b \leftarrow \text{median}_{k \in O_b}(o^S_{a(k)} - \hat{o}_k)$.
      For the next 69 slots, construct y_k^* using Eq. 5:
        translate reference onsets by $-\delta_b$ and clip at zero;
        retain reference durations and pitches; use $\varnothing$ for unmatched notes.
      Append targets interleaved with performance controls to the saved context.
      Mask the 69 retained score triplets from the loss; add the window to $\mathcal{H}_r$.
    end for
  end for
  $\mathcal{D} \leftarrow \mathcal{D} \cup \mathcal{H}_r$ (retain every preceding harvest).
  Initialize π_r from the fixed π_0; anchor to π_0's weights.
  Train π_r on $\mathcal{D}$ for 5,000 updates using Table 3:
    score-token dropout 75% on harvested windows and 0% on clean windows.
end for
return π_2.

B ADDITIONAL COMPARISONS AND CHECKPOINT SELECTION

Table 4 reports paired test-set comparisons between AMT-Q and random initialization, task-specific systems, and the DAgger extension. These complement the aggregate results in Table 1 and the analyses in Sections 5.1 and 5.2.

Table 4: **Paired differences on test**, per performance (59) and over works (mean of 14 work means), in points. IOI at est. scale is IOI F1 after correcting each output for its measured time scale and therefore measures the component of the IOI difference not attributable to time scale. Ref. scale reports outputs at the reference scale for the first versus second system; over works, it reports works on which the first has more versus fewer such outputs. * denotes uncorrected $p < 0.05$: sign-flip test per performance, exact sign-flip test over works, and exact binomial test on discordant outputs for per-performance Ref. scale.

		Strict F1	Scaled F1	IOI F1	IOI at est. scale	Scaled IOI F1	Ref. scale
AMT-Q − random init., same stage 2	perf.	$+5.48^*$	$+4.43^*$	$+6.38^*$	$+3.73^*$	$+2.00^*$	43 / 36
	work	$+5.87^*$	$+4.98^*$	$+6.98^*$	$+6.58^*$	$+2.96$	5 / 2
AMT-Q − Beyer & Dai (2024)	perf.	-0.25	-0.75	$+11.00^*$	$+1.50$	-0.79	43 / 23*
	work	$+0.35$	-2.47	$+9.97$	$+4.96$	$+1.40$	9 / 4
AMT-Q − Zeng et al. (2026)	perf.	$+1.14$	$+0.00$	$+15.76^*$	$+5.56^*$	$+1.94$	43 / 16*
	work	$+2.35$	$+1.24$	$+15.43^*$	$+7.27^*$	$+3.19$	7 / 2*
AMT-Q + DAgger − AMT-Q	perf.	$+0.62$	$+0.41$	-0.88	-2.38^*	-0.59	41 / 43
	work	$+5.09$	$+5.38$	-1.19	-3.54	-1.15	3 / 5

Pretraining and checkpoint selection. The random-initialization arm B8 uses the identical Stage 2 recipe and selection rule as AMT-Q (Table 5). Both runs are evaluated on test every 4k steps for the matched-checkpoint comparison in Section 5.1. The two arms enter Stage 2 from different Stage 1 runs: the random-initialization model uses 20k steps without the anchor, whereas AMT uses step 2,500 of a 20k-step Stage 1 run selected by validation loss.

At Stage 1 alone, where both arms use the same 20k-step recipe and whole-song validation selection, pretraining improves scaled IOI F1 by 3.3 points over test works (10 of 14, $p_{\text{work}} = 0.032$; 2.6 points per performance). The validation comparison has the same sign ($p_{\text{work}} = 0.026$), although strict F1 does not differ significantly. Restricting random-initialization selection to six checkpoints chooses step 40,000 (B8′, 4.25 strict F1 on test); relative to that checkpoint, pretraining is ahead by 4.47 strict-F1 points ($p_{\text{work}} = 0.072$), yielding the same qualitative conclusion.

Table 5: **Random initialization under the Stage 2 recipe and selection rule** (B8; %). Test contains 59 performances of 14 works; validation contains 86 performances of 16 works. Δ is AMT-Q minus B8 per performance. "Works" gives the number of works on which AMT-Q is better/worse. Ref. scale is the number of outputs classified at the reference scale using the local estimate.

	Test						Validation					
Metric	AMT	Rand.	Δ	p_{perf}	works	p_{work}	AMT	Rand.	Δ	p_{perf}	works	p_{work}
Strict F1	8.72	3.23	$+5.48$	$<.001$	11/3	.004	12.13	10.42	$+1.71$	.43	11/5	.62
Scaled F1	10.79	6.36	$+4.43$	$<.001$	11/3	.006	14.39	14.58	-0.19	.92	9/7	.70
IOI F1	78.81	72.43	$+6.38$	$<.001$	11/3	.033	73.52	69.17	$+4.35$	.004	11/5	.14
Scaled IOI F1	82.24	80.24	$+2.00$	.049	9/5	.103	87.72	85.22	$+2.51$	.013	11/5	.13
Global-shift strict	20.77	15.93	$+4.83$	.003	11/2	.022						
Ref. scale	43	36		.14	5/2	.48	57	49		.039	5/2	.25

C METRIC DEFINITIONS

This appendix expands the metric definitions in Section 4. Table 6 reports their response to controlled perturbations.

Table 6: **Response of the evaluation metrics to controlled perturbations.** Known errors are injected into the reference transcriptions of the 59 test performances (mean %; 100 without perturbation). A lost beat moves every subsequent note by one beat; a note-value error moves an onset by $\pm\frac{1}{4}$ beat; timing-noise standard deviation is measured in 10 ms bins. Ref. scale counts outputs measured at the reference scale.

Injected error	Strict F1	Scaled F1	IOI F1	Scaled IOI F1	Strict F1 at est. scale	Ref. scale
Global scale $\times 2$ (in $\mathcal{R}$)	3.8	100	58.9	100	100	0
Global scale $\times 0.95$ (not in $\mathcal{R}$)	2.8	2.8	97.1	97.1	2.8	59
Drift: 1 lost beat	61.3	61.3	99.9	99.9	61.3	59
Note-value errors on 10% of notes	88.8	88.8	87.8	87.8	88.8	59
Timing noise, sd 2 bins	52.4	52.4	54.6	59.7	52.4	59
Missing notes, 10%	91.9	91.9	91.7	91.7	–	–
Scale $\times 2$ + 3 lost beats + 10% note values	3.5	34.6	60.3	87.5	34.6	0

Matching and scale search. Both note lists are shifted so that their earliest onset is zero. Predictions are processed in onset order; each prediction is matched to the nearest-onset unused reference note of the same pitch whose onset differs by at most one bin, and every reference note is used at most once. F1 is the harmonic mean of TP$/|\hat{S}|$ and TP$/|S|$.

For a ratio r, $r \cdot \hat{S}$ multiplies predicted onsets and durations by r and rounds them to integer bins. For the quarter-note baselines, the ratio is applied in their native units before conversion through the beat map. The set $\mathcal{R}$ contains all ratios a/b with

$$a, b \in \left\{\tfrac{1}{2}, 1, \tfrac{3}{2}, 2, 3\right\}$$

beats, corresponding to the eighth-, quarter-, dotted-quarter-, half-, and dotted-half-note beat units annotated by ASAP; after removing duplicates, there are 13 ratios.

For IOI F1, notes are sorted by (onset, pitch). The first note receives interval 0, and each later note receives the onset interval from the preceding note, so chord notes after the first have interval 0. The matcher does not check sequence position, which is why the performance-clock baseline already obtains 50.3% IOI F1.

Time-scale estimates. Output k is attached to performed note k, which either has an aligned reference note or does not. For AMT-Q, this correspondence is exact. The baselines emit one output per note according to their own MIDI readers, which order notes in chords differently and omit 1–5 notes on five performances for Beyer & Dai and 114 notes on one *Ondine* performance for Zeng et al.. In the latter case, output index k may be displaced by as many as 120 notes from its corresponding performed note. Reattaching each baseline output to the performed note for which it was produced, using the same pitch and nearest position, leaves every reference-scale classification unchanged: 23 for Beyer & Dai and 16 for Zeng et al..

The *local* estimate is the median, over 20,000 random pairs of matched notes at most 64 positions apart and separated by 1–4 reference beats, of the ratio between predicted and reference onset differences. This is a subsampled Theil–Sen estimate. The *global* estimate instead uses note pairs separated by at least 2 beats anywhere in the song and therefore also absorbs accumulated drift.

"At the estimated scale" rescales each output once by the ratio in $\mathcal{R}$ nearest in log space to the reciprocal of its local estimate. *Global-shift strict F1* instead shifts the prediction by the single whole-song offset that maximizes matches. We try the three most frequent same-pitch onset differences, smoothed over ± 1 bin. This adjustment forgives one global offset but not accumulated drift.

Reference quantizers. All reference quantizers emit one note per performed note with the performed pitch. The *performance clock* uses the performance onset and duration directly on the 0.5 s-per-beat score clock. The *oracle global tempo* fits

$$b(t) \approx a + ct$$

by least squares to the annotated beats and uses onset $50(a + ct)$ and duration $50cd$. The *oracle beats* uses onset $50b(t)$ and duration

$$50\big(b(t + d) - b(t)\big),$$

where $b(t)$ is interpolated between annotated beat times. Onsets and durations are optionally rounded to a $1/4$- or $1/12$-beat grid. Onsets are clamped at zero because the reference similarly clamps notes preceding the first annotated beat. Table 1 reports the grid for each oracle yielding the highest test strict F1; Table 7 reports all variants.

Table 7: **Reference quantizers under each grid and three time-scale indicators** (test, $n = 59$). Left: F1 (%). Right: number of outputs at the reference scale according to the IOI-optimal ratio, local time-scale estimate, and global time-scale estimate. Rows marked $\star$ appear in Table 1.

				Scaled	Global	Ref. scale		
System	Strict	Scaled	IOI	IOI	shift	IOI ratio	Local	Global
Performance clock	1.13	2.73	50.32	62.03	2.50	8	9	8
Oracle tempo, no grid	3.60	3.65	61.65	63.31	–	29	48	59
Oracle tempo, $\frac{1}{4}$ beat $\star$	9.14	9.39	62.16	77.28	13.65	19	48	59
Oracle tempo, $\frac{1}{12}$ beat	4.57	4.58	64.65	69.35	–	29	48	59
Oracle beats, no grid	55.19	55.19	65.50	65.76	–	53	59	59
Oracle beats, $\frac{1}{4}$ beat	59.76	59.76	64.49	79.30	–	20	59	59
Oracle beats, $\frac{1}{12}$ beat $\star$	61.61	61.61	72.50	73.27	63.00	45	59	59
Beyer & Dai						18	23	22
Zeng et al.						16	16	11
Teacher forcing only						4	6	5
Random init., same stage 2						33	36	32
AMT-Q						45	43	31

Validation of the time-scale estimate. The reference quantizers provide cases in which the relevant time scale is known (Table 7). The oracle-beat outputs are classified at the reference scale on every performance for every grid by both direct slope estimates. The global-tempo oracle is also at the reference scale globally for all 59 performances and locally for 48; the local failures occur when a pianist's local tempo differs from the fitted global tempo by more than the 15% classification band.

By contrast, the IOI-optimal ratio depends on the quantization grid. For the oracle-beat system, it yields reference-scale counts of 53, 20, and 45 for the three grids. Where the IOI-optimal ratio differs from 1, its IOI F1 exceeds the value at ratio 1 by a median of 23.7 points, so this behavior is not caused by tie-breaking.

For the learned systems, which emit on a fine grid, the indicators agree more closely. For AMT-Q, 41 performances are classified at the reference scale by both the IOI indicator and local estimate, four only by the IOI indicator, and two only by the local estimate. The global count for AMT-Q is 31, compared with 43 under the local estimate, consistent with accumulated drift affecting longer-range pairs.

Relation to existing measures. The scaled metrics provide a coarse analogue of the scale invariance in the rhythm-correction cost of Nakamura et al. (2017), which underlies MUSTER's onset error. That cost permits local rescaling, whereas our metrics allow one global ratio from a small musically motivated set, making the selected ratio directly interpretable.

MV2H (McLeod & Steedman, 2018) supports partial evaluations, but its note-value component scores notes with correct within-voice successor links. We do not report that component because AMT-Q predicts no voice assignments; supplying reference voices would incorporate information not predicted by the systems themselves. This limitation does not preclude using other MV2H components.

C.1 MUSTER EVALUATION PROTOCOL

MUSTER's matcher crashes on all seven *Ondine* test performances for every evaluated system, including a reference self-comparison. The baseline evaluation wrapper silently omits crashed pieces. We instead report the same explicitly defined subset for every system: the remaining 52 performances of 13 works.

We convert every system's note-level output and the reference through the same MusicXML writer, using 50 divisions per quarter note, 4/4 meter, one staff, and greedy voice assignment. The export uses `<chord/>`, since the public parser mishandles the valid whitespace form `<chord />`, and corrects underfull-measure export. These are writer-side corrections. The only change to MUSTER's evaluation logic is a one-line fix to offset interpolation. The reference self-comparison in Table 2 still has nonzero extra-note and offset errors after this fix; we report it as a diagnostic rather than an error floor.

Notation also affects the scores independently of note content. In a pilot on one Bach BWV 846 piece, re-notating identical notes changed MUSTER onset error from 0 to 47.60%. This single-piece example motivates controlling the writer; it is not an aggregate result over the test set.

D ABLATIONS ON TEST AND VALIDATION

Figure 5 summarizes the ablations discussed in Section 5.3. Tables 8 and 9 report validation and test results, respectively. The clean-data controls D10–12 match one DAgger round's 5,000 optimizer updates. In E2–E3, each round initializes from and anchors to the preceding round's weights, rather than the fixed Stage 2 model.

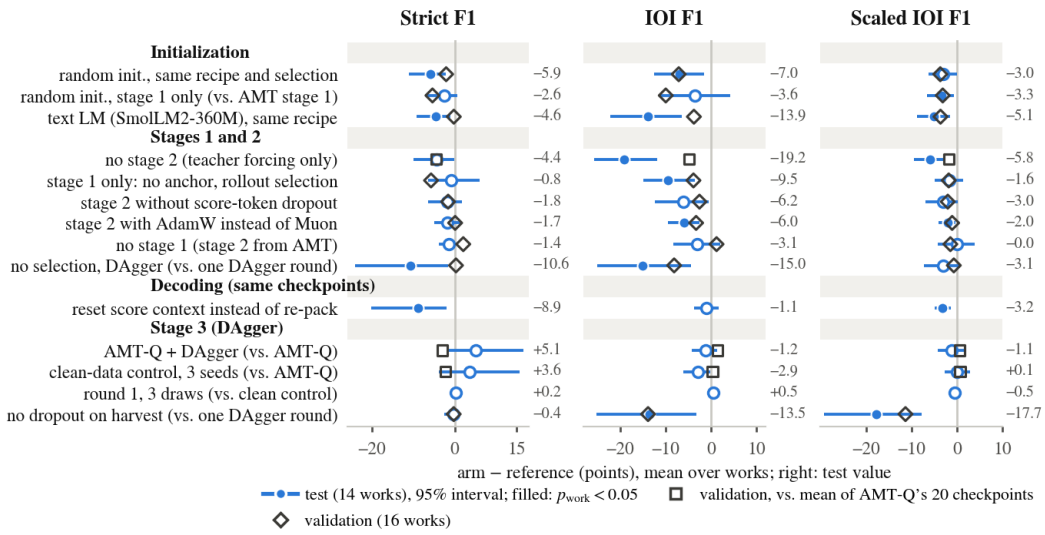


Figure 5: **Ablations of the adaptation pipeline.** Each arm is shown relative to its reference (initialization rows: pretrained AMT under the same recipe; all others: AMT-Q unless stated). Intervals are 95% bootstrap intervals over the 14 test works; filled test markers denote $p_{\text{work}} < 0.05$. Diamonds show the corresponding result over the 16 validation works. Open squares show validation results for Stage 1, selected by validation loss, and arms fine-tuned from AMT-Q, each relative to the mean of AMT-Q's 20 checkpoints. Validation results and per-performance test results appear in Tables 8 and 9, respectively. "Draws" denotes training runs.

Table 8: **Ablation arms on the validation split** (86 performances; means %). Right: arm minus its reference, averaged over the 16 works, with p_{work} in parentheses. References are AMT-Q for the B arms; the mean of AMT-Q's 20 checkpoints for Stage 1 and AMT-Q + DAgger; A1 for A2 and A3; and D1 for the D arms. Ref. scale is the number of outputs at the reference scale according to the local estimate.

	Strict	Scaled	IOI	Sc. IOI	Ref.	Arm − reference, over works (p_{work})		
Arm	F1	F1	F1	F1	scale	strict	IOI	scaled IOI
AMT-Q (B1)	12.13	14.39	73.52	87.72	57	–	–	–
Mean of AMT-Q's 20 checkpoints	9.90	11.90	71.50	86.69				
Stage 1 (B5)	6.56	9.65	61.92	83.02	27	−4.4 (.052)	−4.9 (.24)	−1.7 (.40)
A1: stage 1 only, anchor off	8.59	13.15	69.84	86.01	47	−5.7 (.082)	−3.9 (.21)	−1.8 (.23)
A2: random init., stage 1 only	5.69	9.51	62.70	82.07	44	−5.4 (.58)	−10.0 (.16)	−3.1 (.026)
A3: SmolLM2, stage 1 only	4.98	8.76	63.95	81.80		−6.2 (.37)	−7.3 (.28)	−3.6 (.080)
B2: no score-token dropout	9.03	12.65	69.64	84.90	49	−1.6 (.72)	−2.6 (.35)	−2.0 (.17)
B3: AdamW instead of Muon	9.39	12.93	66.58	85.93	46	+0.1 (.98)	−3.3 (.16)	−1.1 (.30)
B4: no stage 1	11.06	13.06	75.43	88.63	57	+2.1 (.69)	+1.2 (.71)	−1.4 (.28)
B6: SmolLM2, same recipe	9.26	11.25	68.12	82.98		−0.2 (.97)	−3.8 (.21)	−3.6 (.005)
B8: random init., same recipe	10.42	14.58	69.17	85.22	49	−2.1 (.62)	−7.1 (.14)	−3.7 (.13)
AMT-Q + DAgger (run c)	9.51	10.98	76.08	86.64	58	−3.0 (.21)	+1.4 (.67)	+0.6 (.61)
D1: one DAgger round	10.25	12.29	75.95	87.73	60	–	–	–
D2: harvest from stage 1	8.96	12.76	66.89	86.47		+1.6 (.81)	−8.3 (.023)	+1.3 (.41)
D3: from final stage-2 ckpt.	6.20	9.77	61.59	86.24	33	+0.3 (.96)	−8.1 (.056)	−0.7 (.61)
D4: no dropout on harvest	11.48	13.46	65.62	74.71	61	−0.2 (.79)	−13.9 (.003)	−11.3 (.021)
D10: clean data only, seed 1	9.80	11.49	71.26	84.97		−0.9 (.22)	−5.1 (.036)	−0.8 (.39)

Table 9: **All ablation arms** (test, $n = 59$, %). Ref. is the number of outputs for which the IOI-optimal ratio is 1; the local estimate for the main arms is reported in Table 7. Δ and p_{perf} are uncorrected comparisons against the reference named in each block. Each row corresponds to one training run unless otherwise marked.

Row	Configuration	Strict	IOI	Sc. IOI	Ref.	Δ strict (p_{perf})	Δ sc. IOI (p_{perf})
A. Initialization, stage 1 only: 20k steps, whole-song validation selection, anchor off unless stated (reference A1)							
A1	AMT (music-pretrained)	5.30	67.06	80.40	19	–	–
A2	random initialization	4.40	65.44	77.76	24	-0.90 (.27)	-2.64 (.013)
A3	SmolLM2-360M (text embeddings, anchor on)	4.65	56.40	78.15	11	-0.65 (.59)	-2.25 (.052)
B. Stage-2 recipe (reference B1)							
B1	AMT-Q (stage 2)	8.72	78.81	82.24	45	–	–
B2	− score-token dropout, 16k steps	6.21	70.99	79.48	26	-2.51 (.047)	-2.76 (.004)
B3	AdamW instead of Muon, 16k steps	6.43	71.80	81.13	31	-2.29 (.11)	-1.11 (.12)
B4	− stage 1 (stage 2 from AMT)	8.21	79.33	83.45	44	-0.50 (.66)	$+1.22$ (.16)
B5	− stage 2 (= Stage 1)	3.94	61.44	78.11	4	-4.77 (<.001)	-4.12 (<.001)
B6	SmolLM2-360M init., same recipe	3.87	63.65	78.63	14	-4.84 (<.001)	-3.61 (<.001)
B7	B6 + one DAgger round (vs. B6)	4.12	74.23	80.13	39	$+0.24$ (.46)	$+1.50$ (.037)
B8	random init., same recipe and selection	3.23	72.43	80.24	33	-5.48 (<.001)	-2.00 (.049)
B8′	same run, step 40k (best of 6 checkpoints)	4.25	72.25	78.74	27	-4.47 (<.001)	-3.49 (<.001)
C. Decoding: reset the score context in every window instead of re-packing (same checkpoints)							
C1	mean over DAgger rounds 1–3	3.82	75.99	78.68	39–46	-4.77 (<.001)	-4.20 (<.001)
D. One DAgger round from AMT-Q, seed 1 (reference D1)							
D1	AMT-Q + one DAgger round	8.95	76.91	82.70	36	–	–
D2	harvest and train from stage 1	3.45	62.86	78.74	15	-5.50 (<.001)	-3.97 (<.001)
D3	from final stage-2 ckpt. (no selection)	3.19	59.59	79.41	13	-5.75 (<.001)	-3.29 (.006)
D4	− dropout on harvested windows	9.30	60.23	62.47	39	$+0.35$ (.77)	-20.23 (<.001)
D5	− loss mask on retained score slots	8.55	78.67	82.60	40	-0.40 (.66)	-0.11 (.81)
D6	∅ targets left unsupervised	8.86	77.47	82.39	39	-0.08 (.95)	-0.31 (.60)
D7	Muon instead of AdamW	6.77	77.71	82.49	37	-2.18 (.20)	-0.21 (.73)
D8	Muon, heavy weight decay, no anchor	3.58	61.91	70.20	14	-5.36 (<.001)	-12.50 (<.001)
D9	same, anchor on	3.74	64.36	70.83	25	-5.21 (<.001)	-11.87 (<.001)
D10	clean data only, seed 1	7.46	75.24	81.61	37	-1.48 (.17)	-1.09 (.08)
D11	clean data only, seed 2	8.16	77.19	82.68	37	–	–
D12	clean data only, seed 3	7.55	74.76	81.63	35	–	–
DAgger round 2 and clean-data controls (per-performance means over runs or seeds)							
Dc	AMT-Q + DAgger (run c, validation-selected)	9.33	77.93	81.65	41	$+0.62$ (.76)	-0.59 (.47)
Da, Db	runs a, b	9.28, 7.45	78.08, 77.69	83.06, 82.47	39, 38	$+0.56, -1.27$	$+0.82, +0.23$
	three-run mean − AMT-Q	8.69	77.90	82.39	38–41	-0.03 (.99)	$+0.16$ (.83)
	D10–12 (3 seeds) − AMT-Q	7.73	75.73	81.97	35–37	-0.99 (.62)	-0.26 (.70)
	round 1 (3 runs) − D10–12	8.10	77.11	82.63	38–39	$+0.37$ (.52)	$+0.66$ (.090)
E. DAgger variants (reference: the same round of run a); E1 uses no aggregation							
E1	round 3 on $\mathcal{H}_3$ only (7.69)	8.03	77.28	82.52	39	$+0.34$ (.42)	-0.36 (.51)
E2	previous-round init., round 2 (9.28)	8.05	75.45	81.82	34	-1.23 (.23)	-1.24 (.12)
E3	previous-round init., round 3 (7.69)	8.13	77.58	82.08	37	$+0.43$ (.60)	-0.80 (.05)

E EXAMPLE ENGRAVINGS

Figure 6 illustrates the notation produced from the quantized output described in Section 3.2.

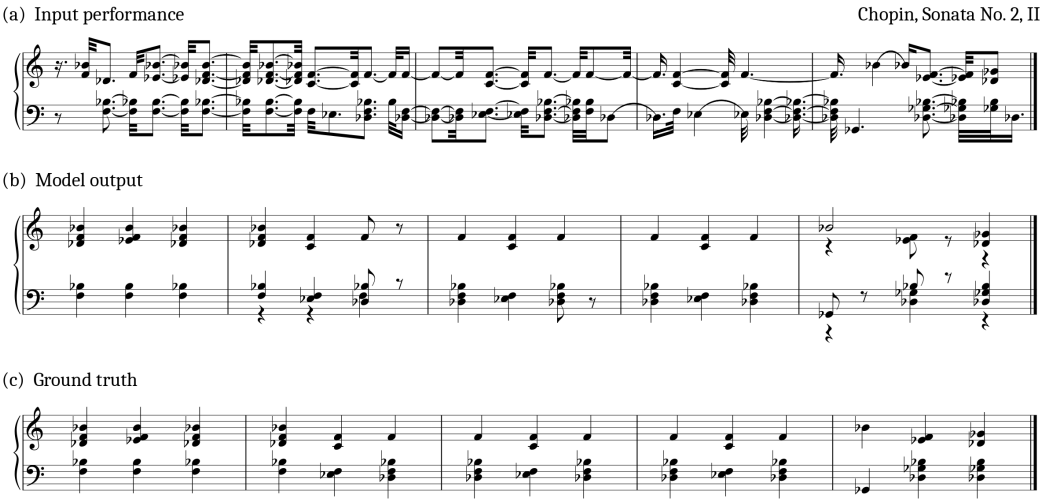


Figure 6: **Example engravings.** Engravings of (a) the raw performance, (b) the quantized performance, and (c) the ground-truth score. Engravings were created from MIDI using `music21`. The quantized output is substantially more regular than the input performance and consequently admits a cleaner sheet-music rendering.