

QUARTETA2S: A MODULAR SYSTEM FOR STAVES-INFORMED QUARTET AUDIO-TO-SCORE TRANSCRIPTION

Di Lu^{1,2,3} Xinran Zhang^{1,2,3} Qingwen Zhou^{1,2,3}
Xiaobing Li^{1,2,3} Maosong Sun^{1,2,3,4,†}

¹ Department of AI Music and Music Information Technology, Central Conservatory of Music

² Key Laboratory of Music and Brain Science,
Central Conservatory of Music, Ministry of Education, China

³ Laboratory of AI Music, Central Conservatory of Music,
Laboratory of Philosophy and Social Sciences, Ministry of Education, China

⁴ Department of Computer Science and Technology, Tsinghua University

ld@ccom.edu.cn, sms@tsinghua.edu.cn

[†]Corresponding author

ABSTRACT

We present QuartetA2S, a submission to the Staves-Informed track of the MIREX 2026 Audio-to-Score Transcription challenge. The system contains two learned stages followed by a deterministic notation backend. Stage 1 combines quartet-adapted YourMT3 transcription with learned four-part assignment to produce part-aware note events. Stage 2 uses a Transformer-based symbolic correction model to refine these events before score construction. Stage 2 is trained on actual Stage 1 predictions paired with clean score-derived targets. The deterministic backend then converts the corrected events and supplied stave metadata into synchronized `**kern` scores. On 6,088 validation excerpts, the final system obtains WER/CER/LER of 0.3592/0.2215/0.6242, with all outputs passing the local structural validator. Local MV2H₄ is 0.8592 on 6,077 successfully evaluated pairs. Processing 12.34h of audio takes 8.51h on one H800 GPU. Official MIREX test results are pending.

1. INTRODUCTION

Audio-to-score (A2S) transcription goes beyond the note events produced by automatic music transcription (AMT). A complete score must also recover part organization, notated durations, rests, meter, and bar structure. In the MIREX 2026 Staves-Informed track, each audio input is accompanied by a `**kern` header that provides stave metadata [1].

Existing A2S work includes direct audio-to-notation models [2,3] and modular pipelines that separate note tran-

scription from score reconstruction [4,5]. QuartetA2S follows the modular approach. It combines quartet-adapted transcription, four-part assignment, symbolic correction, and `**kern` score construction. The symbolic correction model is trained on actual front-end predictions paired with score-derived targets. Our contribution is a quartet-specific modular A2S system that combines part-aware transcription with learned correction of part-assigned transcription outputs.

2. SYSTEM DESCRIPTION

Figure 1 summarizes both training and inference in QuartetA2S. The system contains two learned stages followed by a deterministic notation backend. During training, score-derived supervision is used to adapt the acoustic transcription model, train four-part assignment, and learn symbolic correction from actual Stage 1 predictions. During inference, Stage 1 converts audio into part-aware note events, Stage 2 corrects these events, and the deterministic backend constructs the final synchronized `**kern` score using the supplied stave metadata.

2.1 Stage 1: Acoustic Transcription and Part Assignment

Stage 1 converts quartet audio into part-aware note events. This stage is needed because acoustic transcription provides note content, whereas the Staves-Informed score requires explicit identities for violin I, violin II, viola, and cello.

During training, reference scores are converted into MIDI-like note events with part labels, while score-derived MIDI is rendered into synthetic audio. The acoustic model is adapted to predict note events from the rendered audio. We then run the adapted model on the audio from the corresponding dataset split and match its predictions to the reference events to train the part-assignment model. The two components therefore learn complementary mappings: au-



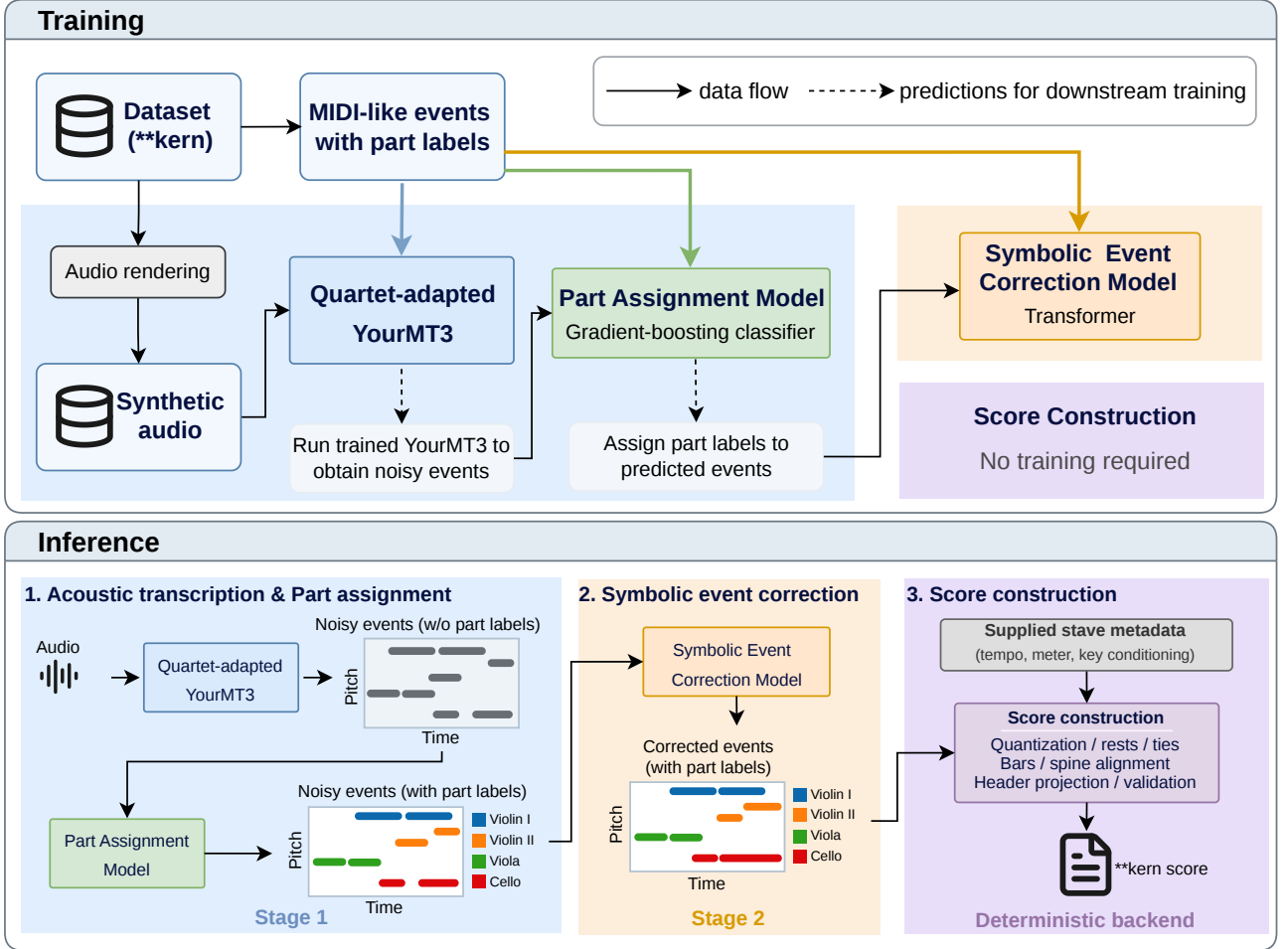


Figure 1. Overview of QuartetA2S training and inference. Stage 1 learns quartet transcription and four-part assignment, while Stage 2 learns symbolic event correction from actual Stage 1 predictions to clean score-derived events. At inference, a deterministic backend converts corrected events and supplied stave metadata into a synchronized `**kern` score.

dio to note events, and predicted note events to score-part identities.

At inference, audio is first transcribed into noisy note events and the part-assignment model assigns each event to one of the four quartet parts. Stage 1 outputs a four-part MIDI-like event sequence for subsequent correction.

2.2 Stage 2: Symbolic Event Correction

Stage 1 predictions may still contain errors in pitch, timing, duration, or part assignment. Stage 2 therefore performs symbolic error correction before score construction.

The correction model is trained on paired event sequences. Its input is the actual output produced by Stage 1 after part assignment, while its target is the corresponding clean event sequence derived from the reference score. The model thus learns to correct the characteristic errors introduced by the upstream transcription and assignment pipeline.

At inference, Stage 2 maps the noisy part-aware events from Stage 1 to a cleaner four-part symbolic sequence. The output remains MIDI-like rather than `**kern` notation, leaving notation construction to the deterministic backend.

2.3 Deterministic Score Construction

Score construction requires no model training. The backend receives the corrected four-part events together with the stave metadata supplied by the task. It quantizes score-time boundaries, inserts rests and ties, reconstructs bars, synchronizes the four parts, and projects the supplied metadata onto the output spines. The result is the final synchronized `**kern` score.

This separation allows the learned stages to focus on uncertain acoustic and symbolic mappings, while deterministic rules handle notation structure that can be derived directly from the corrected events and supplied metadata.

3. IMPLEMENTATION AND TRAINING

Data and rendering. We use the Quartets collection associated with Alfaro-Contreras et al. [2] and retain the supplied training and validation split after duplicate filtering. Score-derived MIDI is rendered with FluidSynth at 16kHz using `FluidR3_GM.sf2`¹. Each excerpt has one rendering at 120 BPM with $\pm 6\%$ tempo variation, velocity

¹ https://member.keymusician.com/Member/FluidR3_GM/

jitter of ± 10 , and 0.5s leading silence. The prepared sets contain 24,333 training and 6,088 validation excerpts, totaling 66.57 and 16.63 audio hours including padding.

Stage 1. We adapt the pretrained YourMT3/YPTF-MoE model [6] from the MT3 family [7]. Official 22.05kHz inputs are resampled to 16kHz. A 0.5s leading silence is added before transcription and removed from predicted timestamps. The selected checkpoint is fine-tuned for 3,000 optimizer steps with learning rate 3×10^{-5} on four H800 GPUs using distributed data parallelism and bfloat16 mixed precision. Pitch-shift augmentation is disabled.

Part assignment uses a gradient-boosting classifier with pitch, duration, local-context, and part-range features. Notes within 40ms are jointly assigned to distinct quartet parts when possible. Predicted notes matched to reference notes provide supervision, yielding 17,274 training notes from 1,912 excerpts and 1,409 validation notes from 253 excerpts.

Stage 2. Supplied tempo q maps onset t seconds to quarter-note beats, $b = tq/60$. Events are represented on a 1/24-beat grid with tokens for part, pitch, onset, duration, velocity, and confidence, preceded by meter and key information. The correction model has six encoder and six decoder layers, hidden size 256, and eight attention heads. It is trained using teacher-forced cross-entropy with label smoothing and decoded greedily. Input and output vocabularies contain 1,680 and 728 tokens.

Stage 2 uses 24,333 training and 6,088 validation pairs. Sources are actual Stage 1 predictions after learned part assignment, and targets are clean events extracted from the reference scores. Training uses AdamW with learning rate 2.2×10^{-4} , weight decay 0.01, dropout 0.12, label smoothing 0.03, batch size one, and eight-step gradient accumulation. A 6% warm-up precedes cosine learning-rate decay. Training runs for 35 epochs, with epoch 30 selected by validation loss.

Score writer and validation. The writer uses only the corrected events and stave metadata supplied by the task. Event boundaries are quantized to 1/24 beat. Durations are written as reciprocal `**kern` tokens, with rests inserted in gaps and ties used across barlines. Header projection restores the supplied spine order and interpretations, while auxiliary spines receive structural or null tokens. Validation checks column counts, durations, termination, equal accumulated spine durations, and bar alignment. Failed correction falls back to Stage 1 events, and conversion failures trigger deterministic repair.

4. DEVELOPMENT EVALUATION

Protocol. We evaluate all 6,088 validation excerpts. Following the task definition [1], WER uses tab/newline-delimited fields, CER uses characters, and LER uses complete lines. Full files, including headers and comments, are retained, with only byte-order marks and line endings normalized. Errors are macro-averaged over excerpts. We also report a local implementation of the four-component MV2H metric based on MV2H v2.2 [8]. MV2H₄ averages Multi-pitch, Voice, Meter, and Note-value scores,

Reconstruction	WER↓	CER↓	LER↓	MV2H ₄ ↑
Final QuartetA2S	.3592	.2215	.6242	.8592

Table 1. Development-set performance of the final QuartetA2S system. WER/CER/LER are computed on all 6,088 validation excerpts, while MV2H₄ is computed on 6,077 successfully evaluated pairs.

omitting the harmonic component. All reported values are development-set measurements.

As shown in Table 1, the final QuartetA2S system obtains WER/CER/LER of 0.3592/0.2215/0.6242 on all 6,088 validation excerpts. MV2H is successfully evaluated for 6,077 excerpts (99.82%), yielding an MV2H₄ score of 0.8592. Ten pairs fail during MV2H conversion, and one exceeds the 600s evaluation limit.

5. RESOURCES AND DISCUSSION

Stage 1 has approximately 47.1M parameters, of which 46.5M are trainable. Stage 1 fine-tuning takes 4.31h on four H800 GPUs. Stage 2 has 12.65M parameters, and takes 5.68h on one H800 GPU. The part-assignment classifier is trained on CPU in 15.08 s and has a serialized size of 2.17 MB. On one dedicated H800, the final pipeline processes 12.34 audio hours in 8.51h.

Current limitations include constant-tempo conversion, finite event context, fixed quantization, canonical quartet part priors, and limited handling of ornaments and changing spine structure. Official MIREX test results are pending. Code: https://github.com/ld04080106/Quartets_a2s_mirex26.

6. ACKNOWLEDGMENTS

This work was supported by the Special Program of the National Natural Science Foundation of China (Grant No. T2341003), and the Advanced Discipline Construction Project of Beijing Universities.

7. REFERENCES

- [1] MIREX, “2026: Audio-to-Score Transcription,” https://music-ir.org/mirex/wiki/2026:Audio-to-Score_Transcription, 2026, accessed: September 26, 2026.
- [2] M. Alfaro-Contreras, A. Ríos-Vila, J. J. Valero-Mas, and J. Calvo-Zaragoza, “A transformer approach for polyphonic audio-to-score transcription,” in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, 2024, pp. 706–710.
- [3] E. Cummins, Z. Huang, A. D’Hooge, Z. Mo, and Y. Ju, “Audio-to-score transcription using pre-trained features, data augmentation, and the new SheetSage-A2S dataset,” arXiv:2608.06165, 2026, version 1, August 2026. [Online]. Available: <https://arxiv.org/abs/2608.06165v1>

- [4] A. Cogliati, D. Temperley, and Z. Duan, “Transcribing human piano performances into music notation,” in *Proc. Int. Society for Music Information Retrieval Conf. (ISMIR)*, 2016, pp. 758–764.
- [5] K. Shibata, E. Nakamura, and K. Yoshii, “Non-local musical statistics as guides for audio-to-score piano transcription,” *Information Sciences*, vol. 566, pp. 262–280, 2021.
- [6] S. Chang, E. Benetos, H. Kirchhoff, and S. Dixon, “YourMT3+: Multi-instrument music transcription with enhanced transformer architectures and cross-dataset STEM augmentation,” in *Proc. IEEE Int. Workshop Machine Learning for Signal Processing (MLSP)*, 2024, pp. 1–6.
- [7] J. Gardner, I. Simon, E. Manilow, C. Hawthorne, and J. Engel, “MT3: Multi-task multitrack music transcription,” in *Proc. Int. Conf. Learning Representations (ICLR)*, 2022. [Online]. Available: <https://arxiv.org/abs/2111.03017>
- [8] A. McLeod and M. Steedman, “Evaluating automatic polyphonic music transcription,” in *Proc. Int. Society for Music Information Retrieval Conf. (ISMIR)*, 2018, pp. 42–49.