

PHRASESPLITGEN: TWO-STAGE PIANO CONTINUATION WITH DEVELOPMENT AND ENDING AWARENESS FOR MIREX 2026

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 PHRASESPLITGEN, a task-specific system for the MIREX 2026 Symbolic Music Generation challenge. Given a 4/4 piano prompt of four measures (optionally with a pickup), the system generates a twelve-measure continuation to form a 16-measure piece. To address the official MIREX evaluation criteria of coherency, structuredness, creativity, and musicality, we first adopt bar-reliable training data to preserve metrical and phrase-level structure. We further employ a two-stage generation strategy, where specialized checkpoints separately produce phrase development (bar 5–12) and phrase ending (bar 13–16). At inference, diversity-oriented stochastic sampling is used to encourage creative outputs, followed by heuristic candidate ranking and diversity-aware selection. The system outputs results directly in the official JSON note-list format for MIREX submission.

1. INTRODUCTION

MIREX 2026 defines Symbolic Music Generation as a Piano Music Continuation task: a system receives a four-measure piano prompt, with an optional pickup measure, and generates twelve further measures to form a sixteen-measure piece. Both prompt and continuation are JSON note lists containing *start*, *pitch*, and *duration*. Time is represented at sixteenth-note resolution and all examples are assumed to be in 4/4 meter [1]. The official subjective evaluation considers coherency to the prompt, structuredness, creativity, and musicality.

We adopt these four criteria not only as evaluation tar-

gets, but also as core design principles across data construction, model training, and inference.

Coherency requires the continuation to stay consistent with the four-measure prompt, which provides the initial thematic context. Autoregressive continuation models are well suited to preserving this dependence [2, 3].

Structuredness requires clear sixteen-measure organization with reliable bar boundaries, metrical positions, and beat hierarchy.

Creativity requires diverse developmental paths beyond deterministic prompt extension. We therefore use diversity-oriented stochastic sampling while conditioning on the preceding context.

Finally, musicality requires smooth musical development across the twelve measures and a convincing sense of phrase ending.

We propose PHRASESPLITGEN, a task-specific piano continuation system designed around these four evaluation dimensions.

2. METHODOLOGY

Our pipeline consists of data preparation, two-stage model specialization, and inference-time candidate selection.

First, we construct bar-aligned sixteen-measure excerpts to preserve metrical structure and meaningful phrase boundaries in the training data.

Second, we train an autoregressive base model for general piano continuation, then fine-tune two specialized checkpoints from this shared base: one for phrase development (bars 5–12) and one for phrase ending (bars 13–16). This split allows the two checkpoints to specialize in development and ending, respectively.

Third, inference employs diversity-oriented sampling together with post-hoc selection, exploring a diverse set of continuations while prioritizing outputs with stronger overall musicality.

Figure 1 summarizes the complete PHRASESPLITGEN pipeline.



© D. Lu, X. Zhang, Q. Zhou, X. Li, and M. Sun. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** D. Lu, X. Zhang, Q. Zhou, X. Li, and M. Sun, “PHRASESPLITGEN: Two-Stage Piano Continuation with Development and Ending Awareness for MIREX 2026”, in *Extended Abstracts for the Late-Breaking Demo Session of the 27th Int. Society for Music Information Retrieval Conf.*, Abu Dhabi, UAE, 2026.

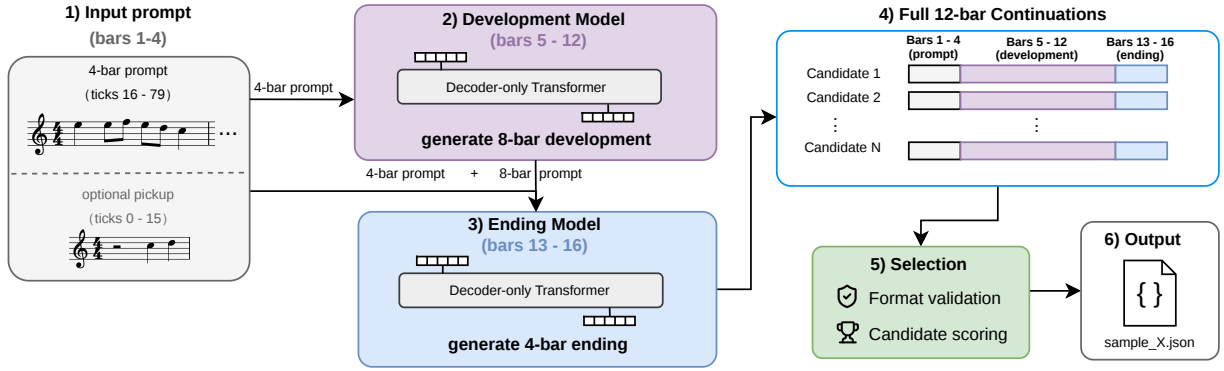


Figure 1. Overview of PHRASESPLITGEN. Given a four-bar piano prompt, the system first generates an eight-bar development with the development model and then completes bars 13–16 with the ending model. Multiple candidates are generated, validated, and scored, followed by diversity-aware selection. After score sorting, the final MIREX-formatted JSON output is produced.

This two-stage factorization relates to prior work that decomposes symbolic piano generation into musically distinct stages [4] and models hierarchical song structure [5]. PHRASESPLITGEN uses lightweight autoregressive event models tailored to the fixed 16-measure MIREX task.

3. DATA AND REPRESENTATION

3.1 Bar-Reliable Training Material

Our data construction prioritizes bar reliability and musical structure. We therefore favor symbolic datasets with explicit measure boundaries and structural information over automatically transcribed audio-to-MIDI data, which may provide less reliable bar and phrase alignment. The resulting corpus combines classical score-derived piano material with filtered pop MIDI.

For classical music, we use score-level MusicXML or `**kern` from PDMX [6], ASAP [7], the DCML Mozart and Beethoven corpora [8, 9], KernScores [10], and the Humdrum keyboard collection [10]. Piano or keyboard parts are retained while preserving their original measure and structural information. For pop music, we use filtered excerpts from POP909 [11], RWC [12], and Lakh [13]. We keep pieces with a long, dense piano accompaniment track, plus a melody track when available. Accompaniment and melody are merged into a single piano stream, and if pitch overlap is persistent, the melody is transposed up by 12 or 24 semitones.

All retained classical and pop excerpts are converted to a uniform 4/4 representation and quantized to a sixteenth-note grid. We extract bar-aligned sixteen-measure windows and discard those containing empty or incomplete bars. As data augmentation, each retained window is transposed within ± 6 semitones whenever all resulting pitches stay within the valid MIDI range.

These sixteen-measure windows form the complete corpus used to train the base continuation model. From this base corpus, we derive two task-specific subsets for subsequent specialization.

The *development subset* uses the first four measures

as context and 5–12 measures as the continuation target. To reduce windows with unstable tonal context, we apply key-based filtering when building this subset. The development subset is further filtered for tonal consistency using Krumhansl–Schmuckler key estimates [14] over the full window and four-bar sub-blocks. The *ending subset* selects sixteen-measure excerpts that reach a phrase or section ending at bar 16. For training, bars 1–12 serve as context to generate target bars 13–16. For pop material, this subset retains only melody-plus-piano excerpts and excludes accompaniment-only outros.

Train/validation splits are computed by a stable hash of `piece_id`, applied before window extraction and transposition. This ensures overlapping excerpts and their transposed variants never cross the split.

Before transposition augmentation, the base corpus contains 78,799 sixteen-measure windows from 5,561 pieces. Table 1 summarizes the resulting dataset sizes.

Table 1. Statistics of the PHRASESPLITGEN training datasets after preprocessing and augmentation.

Training set	Total	Train	Validation
Base dataset	1,024,334	971,385	52,949
Development subset	374,496	355,633	18,863
Ending subset	79,915	75,391	4,524

3.2 Event Encoding

Following REMI’s metrical encoding scheme [2], PHRASESPLITGEN uses BAR, POSITION, PITCH, and DURATION tokens. Each 4/4 bar has sixteen positions, and durations are measured in sixteenth-note units. Velocity, tempo, instrument, and chord tokens are omitted, as they fall outside the MIREX input/output specification. This compact representation enables exact conversion between model output and the official JSON format.

Item	Submission configuration
Representation	BAR / POS / PITCH / DUR
Base architecture	6-layer decoder-only Transformer
Embedding / heads	384 / 6
Context length	2048 tokens
Vocabulary size	158 tokens
Optimizer / w. decay	AdamW / 0.1
LR schedule	Linear warm-up + cosine decay
Dropout	0.15 / 0.10 / 0.10
Initial LR	2×10^{-4} / 3×10^{-4} / 8×10^{-5}
Warm-up steps	3000 / 2000 / 300
Batch / accumulation	8 / 8 (effective batch 64)
Early stopping	Base and ending: patience 8; development: disabled
Final evaluated step	35,000 / 22,000 / 2,250
Effective epochs at final evaluation	7.30 / 14.68 / 5.40
Model parameters	11.49M
Training hardware	One NVIDIA GeForce RTX 2080 Ti
Training time (hour)	6.39 / 3.98 / 0.42

Table 2. Main model and training configuration for the base, development, and ending checkpoints. Values are listed in the order base / development / ending.

4. IMPLEMENTATION

4.1 Model and Training Configuration

All three checkpoints share the same decoder-only Transformer architecture [15]: six layers, six attention heads, 384-dimensional embeddings, and a 2048-token context window. The base model is trained autoregressively on the full sixteen-measure corpus. From this base checkpoint, the development model is fine-tuned for bars 1–4 → 5–12, and the ending model for bars 1–12 → 13–16. Complete model and training hyper-parameters are summarized in Table 2.

4.2 Candidate Generation and Selection

PHRASESPLITGEN first samples candidate continuations for development (bars 5–12), then uses each resulting twelve-bar context to generate candidate endings for bars 13–16. The submission pipeline generates up to 16 development candidates and 6 endings per candidate, producing at most 96 full sequences before final selection. We use IQR-IP sampling [16] with $\rho = 1.5$, top- $k=64$, top- $p=0.96$, and temperatures of 1.05 (development) and 1.0 (ending). IQR-IP reweights the high-probability region after top- k /top- p truncation to reduce token concentration while preserving stochastic sampling.

A generated sequence is parsed back to the sixteenth-note grid before scoring. Candidates with missing or empty bars are strongly down-ranked. A deterministic fallback is used if sampling yields no parsable candidate.

We rank candidates using a weighted combination of heuristic scores, with weights tuned on a held-out validation set:

$$S(c) = \lambda_{\text{pr}} C_{\text{prompt}} + \lambda_{\text{ba}} C_{\text{bars}} + \lambda_{\text{en}} C_{\text{end}} + \lambda_{\text{st}} C_{\text{struct}} + \lambda_{\text{rp}} (1 - P_{\text{rep}}) + \lambda_{\text{rg}} C_{\text{reg}}. \quad (1)$$

- C_{prompt} : prompt matching, comparing note density, polyphony, register, onset histograms, and duration profiles between prompt and continuation.
- C_{bars} : completion score, rewarding exactly twelve generated bars.
- C_{end} : ending quality, favoring sustained final notes, reduced final-bar density, and stable terminal pitch classes when tonal evidence is available.
- C_{struct} : bar-level phrase structure score.
- P_{rep} : repetition penalty combining adjacent-bar similarity and period-2/3 loop detection; period-4 thematic returns are treated as structural repetition and not penalized.
- C_{reg} : register consistency, rewarding preservation of upper-register activity from the prompt.

For our MIREX submission, we empirically set the ranking weights to $\lambda_{\text{pr}}=0.30$, $\lambda_{\text{ba}}=0.18$, $\lambda_{\text{en}}=0.18$, $\lambda_{\text{st}}=0.12$, $\lambda_{\text{rp}}=0.14$, $\lambda_{\text{rg}}=0.08$.

After ranking, we apply a deterministic greedy diversity selector. It keeps the top-scoring candidate and then selects high-ranked candidates that remain sufficiently dissimilar to those already chosen.

5. SUBMISSION INTERFACE

The system entry point is `./generation.sh input.json output_folder n_sample`. The parser supports the full official prompt range, including the optional pickup measure (ticks 0–15) and the four-measure prompt region (up to tick 79). Generated notes start at tick 80 and must fall within the official generation range 80–271 [1].

Post-processing normalizes outputs to the task specification: onsets and durations are integer ticks, pitches stay within MIDI range, and note offsets do not exceed tick 272. Notes are sorted before writing to `sample_XX.json`.

6. CONCLUSION AND REPRODUCIBILITY

PHRASESPLITGEN decomposes the twelve-measure continuation into two conditional subproblems: phrase development (bars 5–12) and phrase ending (bars 13–16). Its core design choices include bar-reliable data construction, development- and ending-aware modeling, and diversity-oriented sampling with post-selection. The submission package includes the `generation.sh` script, frozen model checkpoints, full training and inference configurations, and environment specifications for the submitted system.

7. 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.

8. REFERENCES

- [1] MIREX, “Mirex 2026 symbolic music generation: Piano music continuation,” https://www.music-ir.org/mirex/wiki/2026:Symbolic_Music_Generation, accessed: 2026-08-30.
- [2] Y.-S. Huang and Y.-H. Yang, “Pop music transformer: Beat-based modeling and generation of expressive pop piano compositions,” in *Proceedings of the 28th ACM International Conference on Multimedia*, 2020, pp. 1180–1188.
- [3] Y.-J. Shih, S.-L. Wu, F. Zalkow, M. Müller, and Y.-H. Yang, “Theme transformer: Symbolic music generation with theme-conditioned transformer,” *IEEE Transactions on Multimedia*, 2022.
- [4] S.-L. Wu and Y.-H. Yang, “Compose & embellish: Well-structured piano performance generation via a two-stage approach,” in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 2023, pp. 1–5.
- [5] Z. Wang, L. Min, and G. Xia, “Whole-song hierarchical generation of symbolic music using cascaded diffusion models,” in *The Twelfth International Conference on Learning Representations*, 2024. [Online]. Available: <https://openreview.net/forum?id=sn7CYWyavh>
- [6] P. Long, Z. Novack, T. Berg-Kirkpatrick, and J. McAuley, “PDMX: A large-scale public domain MusicXML dataset for symbolic music processing,” arXiv preprint arXiv:2409.10831, 2024. [Online]. Available: <https://arxiv.org/abs/2409.10831>
- [7] F. Foscari, A. McLeod, P. Rigaux, F. Jacquemard, and M. Sakai, “ASAP: A dataset of aligned scores and performances for piano transcription,” in *Proceedings of the 21st International Society for Music Information Retrieval Conference*, 2020, pp. 534–541.
- [8] J. Hentschel, M. Neuwirth, and M. Rohrmeier, “The annotated mozart sonatas: Score, harmony, and cadence,” *Transactions of the International Society for Music Information Retrieval*, vol. 4, no. 1, pp. 67–80, 2021.
- [9] J. Hentschel, Y. Rammos, M. Neuwirth, F. C. Moss, and M. Rohrmeier, “An annotated corpus of tonal piano music from the long 19th century,” *Empirical Musicology Review*, vol. 18, no. 1, pp. 84–95, 2024.
- [10] C. S. Sapp, “Online database of scores in the humdrum file format,” KernScores, 2005. [Online]. Available: <https://kern.ccarh.org/>
- [11] Z. Wang, K. Chen, J. Jiang, Y. Zhang, M. Xu, S. Dai, X. Gu, and G. Xia, “POP909: A pop-song dataset for music arrangement generation,” in *Proceedings of the 21st International Society for Music Information Retrieval Conference*, 2020, pp. 38–45.
- [12] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka, “RWC music database: Popular, classical, and jazz music databases,” in *Proceedings of the 3rd International Conference on Music Information Retrieval*, 2002, pp. 287–288.
- [13] C. Raffel and D. P. W. Ellis, “Extracting ground-truth information from MIDI files: A MIDIfesto,” in *Proceedings of the 17th International Society for Music Information Retrieval Conference*, 2016, pp. 796–802.
- [14] C. L. Krumhansl, *Cognitive foundations of musical pitch*. Oxford University Press, 1990.
- [15] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” in *Advances in Neural Information Processing Systems 30*, 2017.
- [16] X. Zhang, M. Sun, J. Liu, and X. Li, “Improve the diversity and novelty for open-ended neural text generation via inverse probability weighting,” in *Natural Language Processing and Chinese Computing: 12th CCF International Conference, NLPCC 2023, Part I*, 2023, pp. 774–785.