Getting Motif-ated: Controllable AI Compositions from Injected Motif Prompts

Chao Peter Yang
Yale University
New Haven, CT
chaopeter.yang@yale.edu

Cynthia Rudin
Duke University
Durham, NC

Yue Jiang
Duke University
Durham, NC

Simon Mak
Duke University
Durham, NC

Stephen Ni-Hahn
Chinese University of Hong Kong, Shenzhen
Shenzhen, Guangdong

Abstract

Deep learning has transformed symbolic music generation by borrowing the training paradigms of large language models, with systems such as NotaGen now producing complete, stylistically convincing classical scores from a short prompt. These systems could become powerful creative partners, helping musicians generate endless possibilities. However, current systems expose almost no control handles on the music itself. In principle, control handles could be built into a foundation model trained from scratch, but this is rarely practical without massive amounts of quality annotated data and compute resources. We therefore present MotiGen, a recipe for retrofitting pretrained symbolic music models to use new instruction prompts. MotiGen injects a musical motif as a structured prompt line, reinforces it with a scalar attention bias toward the motif tokens, and learns the association with a two-phase curriculum. First, it learns from focused excerpts cropped around motif occurrences in the training data, then full scores including the motifs. Our experiments show that our model composes with the prompted motif in over 92.3% of generated pieces. Generated pieces using a variety of motifs are included in our sample site: https://motigen-site.github.io/.

1 Introduction

Motifs are the atoms of musical composition. The four notes that open Beethoven's Fifth Symphony are among the most recognizable sounds in Western music, and nearly every bar of the movement grows out of them. Schoenberg called the motif the germ from which coherent musical form develops [Schoenberg, 1967]. When composers have a musical idea, it usually arrives in exactly this form, where a handful of notes with a characteristic contour are developed into full-fledged pieces. Meanwhile, autoregressive language models trained on large score corpora [Huang et al., 2019, Payne, 2019, Wu et al., 2023, Qu et al., 2024] now generate long, well-formed pieces. NotaGen in particular [Wang et al., 2025] showed that the large language model (LLM) paradigm of pretraining, finetuning, and reinforcement learning carries over to musical scores in ABC notation, a textual representation of written music.

However, few models give composers the option to control generation at the motif level, which is the granularity that would allow composers to develop their musical ideas into full compositions. Existing control sits at two extremes, either having coarse controls such as style or genre, or needing fully formed musical material as input. Notagen [Wang et al., 2025], for example, limits the users to coarse inputs such as composer style, period, and instrumentation, offering no control over the

Submitted to 40th Conference on Neural Information Processing Systems (NeurIPS 2026). Do not distribute.

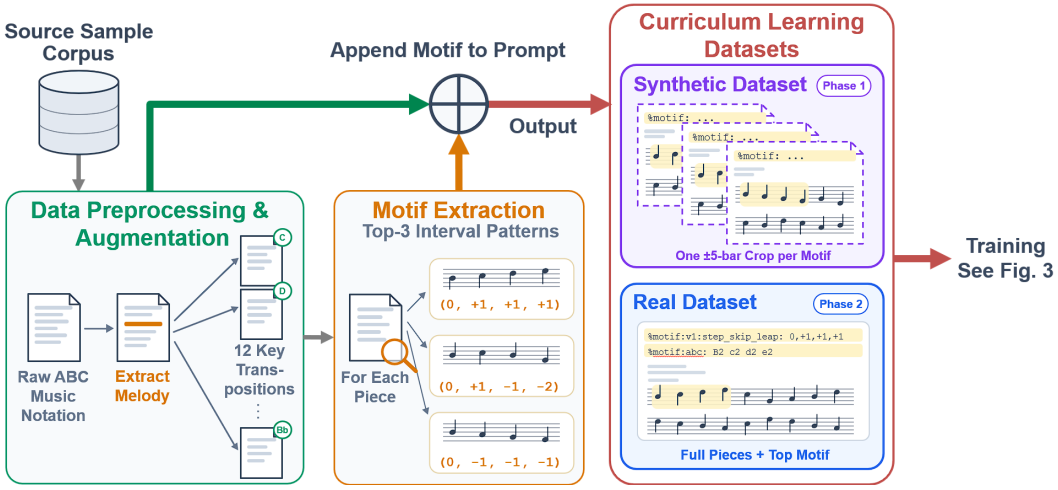


Figure 1: MotiGen data pipeline: each piece is reduced to its melody line, augmented across keys, and mined for its recurring interval patterns; the located occurrences yield the phase-1 focused set of motif-centered crops and the phase-2 of full pieces with `%motif:` annotations.

development of musical material. Alternatively, the user might be expected to provide a concrete, full-fledged melody[Yuan et al., 2024] to harmonize or a bar-level description sequence [von Rütte et al., 2023] that requires fully-composed material.

A motif, in its raw form, is an abstraction that is transposition invariant and may not even have defined intervals, so conditioning on a precise sequence of notes would limit musical generation. In trying to naïvely prompt existing models, we find that pretrained models ignore prompt lines they were not trained for, so models must be tuned such that additional instructions are followed without degrading the backbone's capabilities in generating music. Furthermore, judging whether a generated piece incorporates the motif appropriately requires specialized evaluation methods.

We close this gap with MotiGen. MotiGen enables composers to build melodies and full pieces from a single abstract motif, represented as a transposition-invariant interval-class sequence of steps, skips, and leaps together with their contours. The motif is injected as a structured `%motif:` prompt line in the score header. A scalar *attention bias* at the motif line's key positions keeps the motif salient throughout decoding. A two-phase *curriculum* teaches the model how to incorporate the motif. Phase 1 fine-tunes on focused excerpts cropped from source pieces around located motif occurrences; phase 2 fine-tunes on full scores annotated with their extracted motifs. The result is a model that allows musicians to develop ideas naturally while retaining significant control over the piece's content.

Our novel contributions include: (1) a motif-conditioning method for symbolic music LLMs combining `%motif:` prompt lines with a train- and inference-time attention bias, (2) a focused-crop curriculum that teaches motif grounding from phrases containing each motif and (3) a transposition-invariant, enharmonic-aware motif containment evaluator shared between training annotation and evaluation.

2 Related Work

Generative Symbolic Music Models

Autoregressive Transformers have long been applied to event-token and text-based score representations, from Music Transformer [Huang et al., 2019] and MuseNet [Payne, 2019] to ABC-notation models with bar/patch-aligned hierarchical decoding [Wu et al., 2023, Qu et al., 2024]. NotaGen [Wang et al., 2025] scales this lineage with LLM-style pretraining and preference optimization, and is the backbone for our experiments. None of these models provide control over thematic material.

Control has been investigated for deep learning music models via control tokens [Keskar et al., 2019], learned description sequences [von Rütte et al., 2023], text conditioning [Lu et al., 2023], and hard structural constraints [Herremans and Chew, 2017]. While interpretable probabilistic frameworks give users direct, musically meaningful handles Hahn et al. [2023, 2024], Ni-Hahn et al. [2025], they do not leverage the musical abilities of foundational music LLMs. The closest work to ours is Theme Transformer [Shih et al., 2022], which conditions on a concrete theme involving specific pitches and rhythms and is trained from scratch. We instead condition on an *abstract, transposition-invariant* pattern and retrofit the capability onto a pretrained LLM while preserving its fluency.

Attention and Activation Steering

Classifier-free guidance [Ho and Salimans, 2022], post-hoc attention steering (PASTA) [Zhang et al., 2024], and activation addition [Turner et al., 2023] modify a *frozen* model's computation. Our bias is mechanically similar to PASTA's logit offset but is applied during training as well as inference, so the model learns to exploit the emphasized tokens. Our two-phase schedule is a curriculum [Bengio et al., 2009] in which short focused crops make the motif-content association explicit and direct before full-length scores dilute training with additional musical material.

Motif Discovery in Music Information Retrieval (MIR)

Repeated-pattern discovery is a classical MIR problem, from geometric methods such as SIA/SIATEC [Meredith et al., 2002] and their compression-oriented successors [Collins et al., 2013] to interval-based melodic pattern analysis [Conklin and Anagnostopoulou, 2001] and closed-pattern motivic analysis [Lartillot, 2005]. The task has also been studied directly on audio [Nieto and Farbood, 2014] and remains an active benchmark problem with dedicated datasets [Hsiao et al., 2023] (see Janssen et al. [2013] for a survey). Our contribution is not the discovery algorithm, which, while novel in its representation of the motif, is a simple sliding-window extractor, but rather its use as an effective control signal for generation.

3 Methodology

3.1 Motif Representation and Extraction

A motif must survive transposition and transformation. The opening four notes of Beethoven's Fifth is the same recognizable motif whether it starts on G or F, outlines a major or minor 3rd, or uses quarter notes instead of eighths. We therefore represent a motif not as pitches but as a sequence of *signed interval classes* between consecutive melody notes, after collapsing immediate repetitions (see Figure 2). We measure the distance $d(\cdot, \cdot) \in \mathbb{Z}$ between two notes (n_i, n_j) in terms of diatonic scale steps. For example, $d(A\flat 4, A\flat 4) = 0$, $d(A\flat 3, B\flat 3) = 1$, and $d(C5, F4) = -4$. Intervals are then classified as *steps* when $|d| = 1$, *skips* when $|d| = 2$, and *leaps* when $|d| \geq 3$. We focus on pitch-based motifs and ignore note repetitions, so an interval class for $d = 0$ is not required. Rather, we use 0 as a placeholder for the first note of a motif. For example, `(0,3,-1,-1)` encodes a starting note, a leap up, and two steps down, respectively. The representation is transposition-invariant by construction and deliberately coarse. It captures contour and interval category, which is what most listeners track when they recognize a motif across keys and harmonizations.

To extract the motif from a piece, we use a sliding-window parser. We scan each piece end-to-end using a window length of 4–10 notes, and keep a count of each note sequence's occurrence throughout each piece. We select the most common sequence (weighted by length to mitigate inherent bias towards shorter motifs) to be a defining motif of the piece. For our experiments, we focus on motifs of four notes. Repeats and ornamentation are ignored throughout this process in order to avoid fragmenting motif counts. This extraction technique is used to annotate training data and evaluate generated results (details in Appendix C).

3.2 Conditioning Interface

Novel Prompt Lines: The motif is injected as structured comment lines prepended to the ABC header, replacing NotaGen's period–composer–instrumentation prompt. Concretely, we add `%motif:v1:step_skip_leap: 0,3,-1,-1` as the abstract motif for voice `V:1`, and optionally

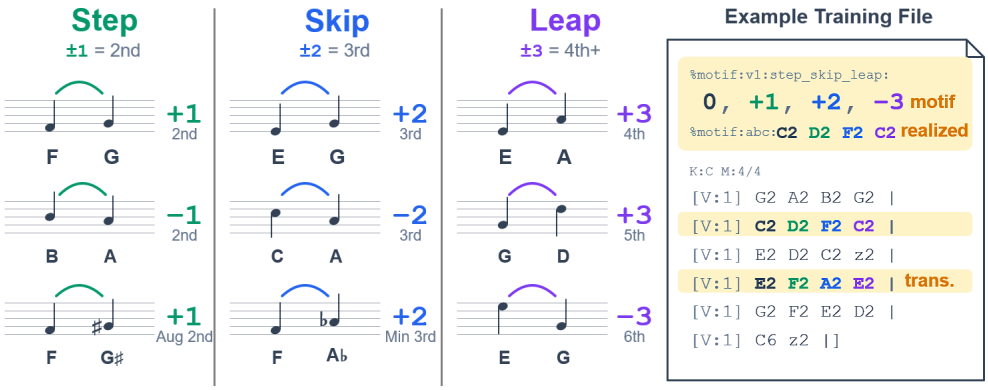


Figure 2: Interval-class encoding. Left: signed letter-step distances map to steps (±1), skips (±2), and leaps (±3). The classification is diatonic, so an augmented second is a step while its enharmonic minor third is a skip. Right: the resulting `%motif:` prompt lines in a training file, with the realization and a transposed recurrence of the same pattern highlighted.

`%motif:abc: G B d c` as one concrete realization (see Figure 2, right). During inference, the model can be run *abstract-only*, where the user provides the abstract motif, and the model will subsequent output the musical phrase containing the actual motif (the realization). Alternatively, the user could also provide the actual motif for MotiGen to use.

Motif Attention Bias: A prompt line alone competes with hundreds of body tokens and its influence decays over a long generation. We therefore add a scalar bias β to the pre-softmax attention logits at the motif prompt's key positions in the patch-level decoder $\alpha_{ij} = q_i^\top k_j/\sqrt{d} + \beta \cdot \mathbf{1}[j \in \mathcal{M}]$, where $\mathcal{M}$ is the set of patch positions belonging to `%motif:` lines. The bias is applied identically at training and inference, so the model learns calibrated use of the emphasized channel rather than having its attention distorted post hoc (Figure 3, bottom). We find that the optimal bias strength for our case is β=4, while β=2 has little effect, and $\beta \geq 8$ collapses generation entirely (more details in Section 4.1).

3.3 Curriculum training

To ease the model into learning new prompt material, we structure training with a two-phase curriculum (overview in Figure 3, top).

Phase 1 (Focused Crops): For every located motif occurrence in the source corpus we build focused cropped training data. The surrounding phrase, cropped to ±5 measures around the occurrence, includes the full original header plus the motif prompt lines. Because these excerpts are real music centered on the motif instead of algorithmically generated samples, the distribution of training data does not drift far from full human-composed pieces. This phase allows the model learn the association between the motif prompts and the generated content of the music more directly.

Phase 2 (Full Pieces): We then fine-tune on full-length scores, each annotated with its extracted motif, with the attention bias active. Here, the model learns how the motif should be present and varied throughout the piece. Curriculum training implementation is detailed in Appendix A.

3.4 Training Datasets and Motivic Analysis

We build on NotaGen [Wang et al., 2025], pretrained on ∼1.6M ABC-notation pieces and fine-tuned by its authors on ∼9k classical scores with period–composer–instrumentation prompts; all of our training modifies this checkpoint. Our source-data pool combines the OpenScore Lieder corpus [Gotham and Jonas, 2022] (∼1.3k Romantic-era songs, from which we take the `V:1` vocal line) with the Irishman corpus [Wu, 2023] of 216k+ monophonic folk tunes, each wrapped into a single-voice score so both sources share one format, these pieces are then augmented across 12 keys as shown in Figure 1. The focused cropped phase-1 set consists of ∼650k+ motif-centered crops built from the same pool (Section 3.3). Given the available dataset, the model will be trained to be monophonic only. In future works, when comparable or larger homophonic datasets becomes available, we plan to

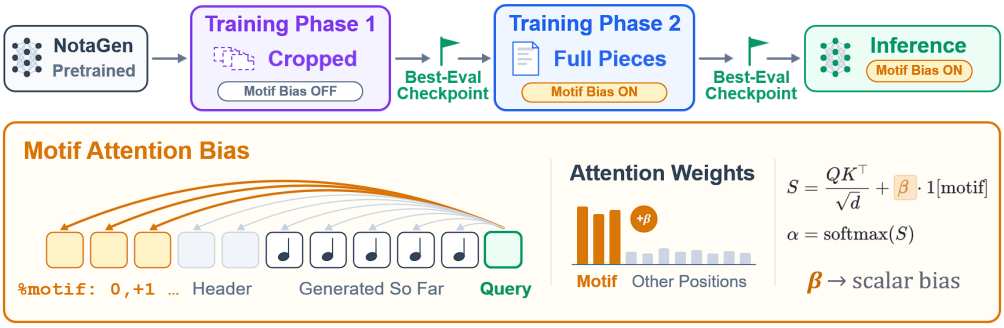


Figure 3: Two-phase curriculum with phase-matched evaluation (top) and an illustration of the motif attention bias (bottom), where a scalar β added to the pre-softmax attention logits at the `%motif:` key positions, identically at training and inference.

expand the model to generate multiple voices as well. An exploratory analysis of the extracted motifs motivates our evaluation design: the source pool contains 210 distinct length-4 motif patterns whose frequencies are strongly skewed. Stepwise descent `(0,-1,-1,-1)` occurs 62,356 times, the rank-27 pattern appears 1,083 times, and pure leap chains such as `(0,3,3,3)` appearing twice in the entire pull (details in Appendix D). We therefore stratify our evaluation based motif frequency.

4 Experiments

To evaluate MotiGen, our experiments investigates how well MotiGen can reliably compose music that organically integrates a prompted motif. We break this down into the following manageable questions: (1) To what extent do our methods increase the model's performance? (Section 4.1), (2) how does performance scale with full-data quantity (Appendix B), and (3) Does MotiGen incorporate the motif musically into the generated piece? We provide a case study of results in Section 4.2, and plan for larger scale human evaluation in future works.

For each (model, motif) pair we generate 50 pieces with the abstract-only prompt and fixed decoding (top-k=9, top-p=0.9, temperature 1.2), matching NotaGen's recommended sampling settings. The resulting samples are then evaluated using two metrics, one is *body containment*, which asks whether the generated melody contains the target pattern anywhere, according to the motif extractor of Section 3.1. Another is *realization faithfulness*, which asks whether the model's own emitted `%motif:abc` line realizes the target. Their 2×2 crosstab separates a model that understands a motif from one that further composes with it. The motifs used in experiments span the training distribution, namely we draw a stratified sample consisting of eight from the top-10 (band A) most frequent patterns, eight mid-distribution patterns (ranks 11–40 of 210, band B), and eight from ranks 41 and rarer (band C).

4.1 Ablations

We ablate attention bias and curriculum training on the same protocol (Table 1). To test the efficacy of attention bias, we perform curriculum training with bias β=0 during both training and inference. To examine the benefits of curriculum training, we remove the cropped-data training phase, fine-tuning only on full pieces. Lastly, we judge performance after ablated both attention bias and cropped-data training phase.

Results provide strong evidence that our full recipe is effective based on all three metrics. Furthermore, each ablated component is shown to positively contribute to motivic composition. Removing the attention bias costs the most while removing the cropped training phase has a milder effect on mid-distribution motifs. The cropped phase matters far more for motifs from the distribution tail and at small corpus scale (Appendix B). Interestingly, plain fine-tuning on the annotated corpus already reaches 79.4% containment against a 35.9% base rate. So it seems the prompt format itself is learnable from data alone, and the motif bias and curriculum training are complementary rather than additive; the cropped curriculum with the bias removed performs no better than plain fine-tuning, whereas together they lead on every metric.

Configuration	Containment	Faithfulness	Motifs / Measure
MotiGen (bias β=4 + curriculum)	**92.3%** [90.7, 93.7]	**93.0%** [91.4, 94.3]	**0.25**
– attention bias	77.8% [75.4, 80.1]	75.8% [73.3, 78.2]	0.18
– cropped phase	86.7% [84.6, 88.5]	87.2% [85.2, 89.0]	0.22
– both	79.4% [77.0, 81.6]	76.1% [73.6, 78.4]	0.19
Base rate (no motif prompt)	36.8% [34.1, 39.5]	—	0.06

Table 1: Ablation of MotiGen components: body containment (the piece contains the motif at least once) and realization faithfulness, averaged over the 24 stratified sample motifs at n=50 per motif. β=4 configurations use training-matched bias application at inference. Bracketed values are 95% Wilson score intervals on the pooled counts (n=1200). See Appendix C for the definitions of Containment and Faithfulness.

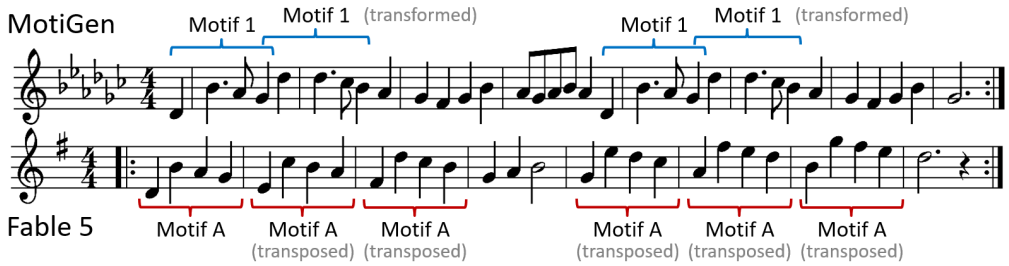


Figure 4: Excerpts from MotiGen (upper) and Fable 5 (lower) when prompted to generate a melody with the motif of a leap up, and two steps down.

4.2 Case Study

In order to get a better understanding of the subjective, qualitative qualities of the music generated by MotiGen, both by itself, and in context of other methods, we provide a case study to examine a piece generated by MotiGen. We prompt the model with a somewhat common, mid-distribution motif of (0, 3, -1, -1), which is a leap up, and two steps down, and allow the model to generate the realization, key signature, and time signature freely.

Since no model to our knowledge generates music based on the same level of abstraction, we prompt a state-of-the-art general-purpose LLM, Fable 5 [Anthropic, 2026], with the following prompt and default settings to produce a comparable piece of music:

```
Generate a melody with a motif that goes a leap up
and two steps down. Feel free to use any key signature
or time signature, output in ABC notation.
```

Figure 4 shows the melody generated by MotiGen not only includes the motif as expected, but also includes interesting and diverse rhythms, as opposed to the simple quarter note sequence as created by the Fable 5 model. The realization of the motif is also more diverse in the sample generated by MotiGen, as it includes both a leap with 6th (measure 1-2), and a 5th (measure 2), while all the motifs in the sample by Fable 5 are all a 6th up and two steps down. Not to mention, the Fable 5 composition contains only the motif and nothing else other than the small cadence in measure 4, while MotiGen has ample connecting materials between occurrences of the motifs.

5 Conclusion

We introduce MotiGen, a recipe for retrofitting a thematic-control channel onto a pretrained symbolic music model by representing the motif as a transposition-invariant interval-class pattern, injecting it as a structured prompt line, reinforcing it with a matched train/inference attention bias, and training it with a focused-crop curriculum. Our resulting model composes convincing music with motivic material organically integrated into the texture rather than simply appended to the beginning. The same recipe could carry other musical instructions: harmonic progressions, bass lines, rhythmic cells, formal plans.

References

Anthropic. Claude fable 5. Large language model, June 2026. URL `https://www.anthropic.com/news/claude-fable-5-mythos-5`. Version `claude-fable-5`, released 9 June 2026.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th International Conference on Machine Learning*, pages 41–48, 2009.

Tom Collins, Andreas Arzt, Sebastian Flossmann, and Gerhard Widmer. SIARCT-CFP: Improving precision and the discovery of inexact musical patterns in point-set representations. In *Proceedings of the 14th International Society for Music Information Retrieval Conference (ISMIR)*, pages 549–554, 2013.

Darrell Conklin and Christina Anagnostopoulou. Representation and discovery of multiple viewpoint patterns. *Proceedings of the International Computer Music Conference*, 2001. TODO: verify venue/pages, or swap for another interval-class pattern-discovery reference.

Mark Gotham and Peter Jonas. The OpenScore Lieder corpus. In *Music Encoding Conference Proceedings 2021*, 2022. Corpus at `https://musescore.com/openscore-lieder-corpus`.

Stephen Hahn, Rico Zhu, Simon Mak, Cynthia Rudin, and Yue Jiang. An interpretable, flexible, and interactive probabilistic framework for melody generation. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 4089–4099, 2023.

Stephen Hahn, Jerry Yin, Rico Zhu, Weihan Xu, Yue Jiang, Simon Mak, and Cynthia Rudin. SentHYMNent: An interpretable and sentiment-driven model for algorithmic melody harmonization. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 5050–5060, 2024.

Dorien Herremans and Elaine Chew. MorpheuS: Generating structured music with constrained patterns and tension. *IEEE Transactions on Affective Computing*, 10(4):510–523, 2017.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

Yo-Wei Hsiao, Tzu-Yun Hung, Tsung-Ping Chen, and Li Su. BPS-Motif: A dataset for repeated pattern discovery of polyphonic symbolic music. In *Proceedings of the 24th International Society for Music Information Retrieval Conference (ISMIR)*, pages 281–288, 2023.

Cheng-Zhi Anna Huang, Ashish Vaswani, Jakob Uszkoreit, Noam Shazeer, Ian Simon, Curtis Hawthorne, 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*, 2019.

Berit Janssen, W. Bas de Haas, Anja Volk, and Peter van Kranenburg. Discovering repeated patterns in music: State of knowledge, challenges, perspectives. In *Proceedings of the 10th International Symposium on Computer Music Multidisciplinary Research (CMMR)*, pages 225–240, 2013.

Nitish Shirish Keskar, Bryan McCann, Lav R. Varshney, Caiming Xiong, and Richard Socher. CTRL: A conditional transformer language model for controllable generation. *arXiv preprint arXiv:1909.05858*, 2019.

Olivier Lartillot. Efficient extraction of closed motivic patterns in multi-dimensional symbolic representations of music. In *Proceedings of the 6th International Conference on Music Information Retrieval (ISMIR)*, pages 191–198, 2005.

Peiling Lu, Xin Xu, Chenfei Kang, Botao Yu, Chengyi Xing, Xu Tan, and Jiang Bian. MuseCoco: Generating symbolic music from text. *arXiv preprint arXiv:2306.00110*, 2023.

David Meredith, Kjell Lemström, and Geraint A. Wiggins. Algorithms for discovering repeated patterns in multidimensional representations of polyphonic music. *Journal of New Music Research*, 31(4):321–345, 2002.

Stephen Ni-Hahn, Chao Péter Yang, Mingchen Ma, Cynthia Rudin, Simon Mak, and Yue Jiang. ProGress: Structured music generation via graph diffusion and hierarchical music analysis. In *Advances in Neural Information Processing Systems*, 2025.

Oriol Nieto and Morwaread M. Farbood. Identifying polyphonic patterns from audio recordings using music segmentation techniques. In *Proceedings of the 15th International Society for Music Information Retrieval Conference (ISMIR)*, pages 411–416, 2014.

Christine Payne. MuseNet. OpenAI, `https://openai.com/blog/musenet`, 2019.

Xingwei Qu, Yuelin Bai, Yinghao Ma, Ziya Zhou, Ka Man Lo, Jiaheng Liu, Ruibin Yuan, Lejun Min, Xueling Liu, Tianyu Zhang, Xinrun 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 Chen, Wenhao Huang, Wenhu Chen, Jie Fu, and Ge Zhang. MuPT: A generative symbolic music pretrained transformer. *arXiv preprint arXiv:2404.06393*, 2024. Check for a published version before submission.

Arnold Schoenberg. *Fundamentals of Musical Composition*. Faber & Faber, London, 1967.

Yi-Jen Shih, Shih-Lun Wu, Frank Zalkow, Meinard Müller, and Yi-Hsuan Yang. Theme transformer: Symbolic music generation with theme-conditioned transformer. *IEEE Transactions on Multimedia*, 25:3495–3508, 2022.

Alexander Matt Turner, Lisa Thiergart, David Udell, Gavin Leech, Ulisse Mini, and Monte MacDiarmid. Activation addition: Steering language models without optimization. *arXiv preprint arXiv:2308.10248*, 2023.

Dimitri von Rütte, Luca Biggio, Yannic Kilcher, and Thomas Hofmann. FIGARO: Generating symbolic music with fine-grained artistic control. In *International Conference on Learning Representations*, 2023.

Yashan Wang, Shangda Wu, Jianhuai Hu, Xingjian Du, Yueqi Peng, Yongxin Huang, Shuai Fan, Xiaobing Li, Feng Yu, and Maosong Sun. NotaGen: Advancing musicality in symbolic music generation with large language model training paradigms. *arXiv preprint arXiv:2502.18008*, 2025.

Shangda Wu. The Irishman dataset. `https://huggingface.co/datasets/sander-wood/irishman`, 2023. ABC-notation corpus of $\sim$216k Irish folk tunes.

Shangda Wu, Xiaobing Li, Feng Yu, and Maosong Sun. TunesFormer: Forming Irish tunes with control codes by bar patching. *arXiv preprint arXiv:2301.02884*, 2023.

Ruibin Yuan, Hanfeng Lin, Yi Wang, Zeyue Tian, Shangda Wu, Tianhao Shen, Ge Zhang, Yuhang Wu, Cong Liu, Ziya Zhou, Ziyang Ma, Liumeng Xue, Ziyu Wang, Qin Liu, Tianyu Zheng, Yizhi Li, Yinghao Ma, Yiming Liang, Xiaowei Chi, Ruibo Liu, Zili Wang, Pengfei Li, Jingcheng Wu, Chenghua Lin, Qifeng Liu, Tao Jiang, Wenhao Huang, Wenhu Chen, Emmanouil Benetos, Jie Fu, Gus Xia, Roger Dannenberg, Wei Xue, Shiyin Kang, and Yike Guo. Chatmusician: Understanding and generating music intrinsically with llm, 2024. URL `https://arxiv.org/abs/2402.16153`.

Qingru Zhang, Chandan Singh, Liyuan Liu, Xiaodong Liu, Bin Yu, Jianfeng Gao, and Tuo Zhao. Tell your model where to attend: Post-hoc attention steering for LLMs. In *International Conference on Learning Representations*, 2024.

A Implementation Details

Our code is available on Github[1]. For curriculum learning, checkpoints are selected by early stopping on held-out loss for up to 10 epochs, and evaluation is phase-matched. Each phase is evaluated on data from its own distribution and in its own bias configuration. Evaluating a bias-trained model with the bias off understates its quality and distorts the training curve; matching removes a spurious cross-phase jump that would otherwise suggest phase 2 begins by "undoing" phase 1. Figure 3 summarizes the schedule.

[1] `https://github.com/cpyang123/motigen`

All fine-tuning starts from the public NotaGen-large checkpoint (20-layer patch-level encoder, 6-layer character-level decoder, hidden size 1280, patch length 1024, patch size 16) in NotaGen's patch-streaming regime, using AdamW with learning rate 10^{-5} and batch size 1. Every training run fits on a single NVIDIA RTX A6000 (48 GB); the reported experiments (the ablation grid of Table 1, the scaling grid of Appendix B, and their 50-piece-per-motif evaluation sweeps) total roughly under a GPU-month on this hardware.

B Data scaling improves motif adoption

Beyond the main ablations, we ask whether motif adoption improves with more training data, holding the recipe fixed. To do this, we train all the model variants, ablated models and the full model, on two subsets of the training data. Specifically, a 10k subset that is an order of magnitude smaller than the 200k+ full dataset, and a 1k dataset that is another order of magnitude smaller still. Additionally, in order evaluate the models performance on motifs of different obscurity, we do the same stratified sample of 8 motifs from the top 10 motifs in the full dataset, middle 11-40 motifs, and the tail 41+ motifs, corresponding to A, B, and C respectively as in Table 2. Each motif is sampled 50 times, and the score calculated as an average over all the generated pieces. The Full Training block reports the underlying component results of Table 1.

B.1 Full per-motif results

Closely examining Table 2 reveals quickly that the full advantage of the full training pipeline is greatest when we have few data points, second when we have relatively abundant data, and worse when the paucity of data greatly hinders generalization across all configurations.

Utilizing the entirety of the training regiment almost unanimously provides better results when trained over the 10k subset and the full training dataset. This is especially prominent when looking at the generations using motifs from the tail-end of the distribution, those of rank C. When prompted with these motifs, the full training recipe produces almost consistently a 10% improvement in both containment and faithfulness over the next best ablation configuration when trained on either the full and the 10k dataset. This could indicate a better internal understanding of the motivic structure and its relation to the generated piece, leading to better generalization on sparser data.

C Motif extraction details

Pitch representation

The key-signature-aware ABC parser tracks every note n in two parallel coordinates: its *chromatic* pitch $\pi(n) \in \mathbb{Z}$ (semitones, with the key signature and bar-persistent explicit accidentals applied) and its *diatonic* staff position $\delta(n) = \ell(n) + 7\,o(n) \in \mathbb{Z}$, where $\ell(n) \in \{0, \ldots, 6\}$ is the letter index (C = 0, D = 1, …, B = 6) and $o(n)$ the octave. δ depends only on the *spelling* of the note, which is what lets the classifier distinguish enharmonically equivalent intervals.

Repeat collapse

Given the note sequence $p_1, \ldots, p_N$ of a voice, immediate repetitions are removed by keeping only the first note of each equal-pitch run: note i is kept iff $i = 1$ or $\pi(p_i) \neq \pi(p_{i-1})$. A motif is a contour of *distinct* successive notes, so a realization that restrikes a note inside the motif span (a tied or repeated tone) still counts as containing it.

Interval classification

Write $q_1, \ldots, q_n$ for the collapsed sequence, and for consecutive notes let $d_t = \delta(q_t) - \delta(q_{t-1})$ (signed letter-steps) and $s_t = \pi(q_t) - \pi(q_{t-1})$ (signed semitones). The interval class is

$$c_t = \begin{cases} \operatorname{sign}(s_t) & \text{if } d_t = 0,\ s_t \neq 0, \\ \operatorname{sign}(d_t) \cdot g(|d_t|) & \text{otherwise,} \end{cases} \qquad g(a) = \begin{cases} 0 & a = 0, \\ 1 & a = 1 \text{ (step)}, \\ 2 & a = 2 \text{ (skip)}, \\ 3 & a \geq 3 \text{ (leap)}, \end{cases} \tag{1}$$

	Containment (%)			Faithfulness (%)		
Configuration	A	B	C	A	B	C
1*k source training pieces*						
MotiGen (full recipe)	**55.0** [50, 60]	27.2 [23, 32]	19.0 [15, 23]	**20.2** [17, 24]	4.2 [3, 7]	1.5 [1, 3]
− attention bias	54.0 [49, 59]	30.2 [26, 35]	**20.5** [17, 25]	14.2 [11, 18]	**4.8** [3, 7]	**3.0** [2, 5]
− cropped phase	52.8 [48, 58]	29.0 [25, 34]	16.5 [13, 20]	15.0 [12, 19]	4.2 [3, 7]	1.0 [0, 3]
− both	53.5 [49, 58]	**31.5** [27, 36]	16.0 [13, 20]	10.5 [8, 14]	3.2 [2, 5]	1.2 [1, 3]
Baseline	47.0 [42, 52]	30.2 [26, 35]	14.5 [11, 18]	—	—	—
10*k source training pieces*						
MotiGen (full recipe)	**84.8** [81, 88]	**64.5** [60, 69]	**53.0** [48, 58]	**84.5** [81, 88]	**68.0** [63, 72]	**59.2** [54, 64]
− attention bias	73.5 [69, 78]	61.0 [56, 66]	43.5 [39, 48]	58.2 [53, 63]	49.8 [45, 55]	38.0 [33, 43]
− cropped phase	68.5 [64, 73]	39.2 [35, 44]	21.2 [18, 26]	38.8 [34, 44]	21.0 [17, 25]	4.2 [3, 7]
− both	60.2 [55, 65]	46.8 [42, 52]	20.2 [17, 24]	21.8 [18, 26]	10.5 [8, 14]	3.2 [2, 5]
Baseline	47.8 [43, 53]	32.5 [28, 37]	16.0 [13, 20]	—	—	—
Full Training						
MotiGen (full recipe)	**96.2** [94, 98]	**94.0** [91, 96]	**86.8** [83, 90]	**95.2** [93, 97]	**96.8** [95, 98]	**87.0** [83, 90]
− attention bias	89.8 [86, 92]	86.5 [83, 90]	57.2 [52, 62]	85.5 [82, 89]	86.8 [83, 90]	55.2 [50, 60]
− cropped phase	95.8 [93, 97]	87.0 [83, 90]	77.2 [73, 81]	95.0 [92, 97]	87.2 [84, 90]	79.5 [75, 83]
− both	87.8 [84, 91]	84.0 [80, 87]	66.5 [62, 71]	82.2 [78, 86]	80.8 [77, 84]	65.2 [60, 70]
Baseline	53.2 [48, 58]	38.0 [33, 43]	19.0 [15, 23]	—	—	—

Table 2: Training recipe crossed with source-data scale, on 24 motifs stratified by corpus frequency (A: ranks 1–10; B: 11–40; C: 41–210; n=50 generations per motif, 8 motifs per stratum, excluding erroneous generations. The baseline row is the unprompted Notagen model i.e. the rate at which the pattern appears by chance in its own idiom. Schedules are matched across scales; β=4 is used for attention bias. Bracketed values are 95% Wilson score intervals on the pooled counts

where the first case counts a chromatic alteration of the same letter (e.g. F to F♯: $s_t \neq 0$ but $d_t = 0$) as a step, since the notes are distinct pitches; an interior $c_t = 0$ cannot occur because identical pitches are removed by the repeat collapse. Classification uses the letter-step distance d_t, so enharmonic spelling is honored: C to D♯ (s_t=3, d_t=1) is a step, while C to E♭ (s_t=3, d_t=2) is a skip. A window of k collapsed notes $(q_i, \ldots, q_{i+k-1})$ maps to the pattern $m = (0, c_{i+1}, \ldots, c_{i+k-1})$, the leading 0 anchoring the first note. When spelling information is unavailable (plain semitone input, e.g. MIDI-derived), d_t is approximated from s_t by a nearest-interval table: $|s_t| \bmod 12 \in \{1, 2\} \mapsto 1$, $\{3, 4\} \mapsto 2$, $\{5\} \mapsto 3$, $\{6, 7\} \mapsto 4$, $\{8, 9\} \mapsto 5$, $\{10, 11\} \mapsto 6$, plus 7 per octave.

Since this method is interval-based, it is transposition-invariant. Transposing a passage adds a constant $(\Delta\delta, \Delta\pi)$ to every note's coordinates, leaving all differences (d_t, s_t) — and hence the pattern m — unchanged. Empirically, a sweep over 2,353 pieces × 12 key transpositions produced identical extracted motifs in all but 29 cases, each traced to inconsistencies in the source encodings rather than to the parser.

Extraction and containment

Algorithm 1 gives the per-piece extraction. Containment checking at evaluation time reuses the same parse, collapse, and classification (Eq. 1): a generated melody *contains* target m iff some length-$|m|$ window of its collapsed note sequence maps to m.

Evaluation metrics

Body containment asks whether the model *composed with* the requested motif. The generated melody is parsed, collapsed, and classified exactly as in training annotation (Eq. 1), and the piece counts iff some length-$|m|$ window maps to the target pattern m. Any realization of the contour, in any key and anywhere in the piece, counts; the prompt lines are excluded from the search. We report the fraction of n=50 generations per motif that pass.

Realization faithfulness asks whether the model *understood* the request. Given only the abstract pattern, the model opens its output with a `%motif:abc:` line stating the notes it intends to use. Faithfulness parses that line alone with the same pipeline and checks that it maps to the target; a

Algorithm 1 Per-piece motif extraction (window sweep 4–10, stride 1)

Require: ABC voice V; window range $w_{\min}$=4, $w_{\max}$=10
Ensure: motif pattern m^* and one concrete realization

$(p_1, \ldots, p_N) \leftarrow \textsc{ParseABC}(V)$ ▷ per-note (π, δ) and bar index
$(q_1, \ldots, q_n) \leftarrow$ collapse immediate equal-pitch runs of p
for $w = w_{\min}$ **to** $w_{\max}$ **do**
  for $i = 1$ **to** $n - w + 1$ **do**
    $m_i \leftarrow (0, c_{i+1}, \ldots, c_{i+w-1})$ by Eq. 1
    $\text{count}_w[m_i] \leftarrow \text{count}_w[m_i] + 1$; record occurrence (note span, bar)
  end for
  $C_w \leftarrow \max_m \text{count}_w[m]$
end for
$w^* \leftarrow \arg\max_w C_w$ (smallest maximizing w); $m^* \leftarrow \arg\max_m \text{count}_{w^*}[m]$
return m^* and the bar-aligned ABC snippet of its first occurrence

missing or malformed line counts as unfaithful. Since faithfulness reads only this header line and containment reads only the tune body, the two are independent, and their cross tabulation separates a model that merely echoes the instruction from one that uses it.

D Training Data Motivic Analysis

Figure 5 shows the rank frequency distribution of the extracted motif annotations over the full training corpus (all Lieder plus the full Irishman set). The distribution is heavily skewed: 210 distinct length-4 patterns occur across 217k annotated pieces, but the two stepwise runs `(0,-1,-1,-1)` and `(0,1,1,1)` alone annotate 48% of them, and counts fall roughly geometrically through the remaining ranks. This skew is what motivates the stratified evaluation of Appendix B: results averaged over unstratified motifs would be dominated by patterns the model has seen tens of thousands of times, hiding exactly the regime where conditioning is hard. The shaded bands mark the head, mid, and tail frequency regimes from which the evaluation motifs are drawn.

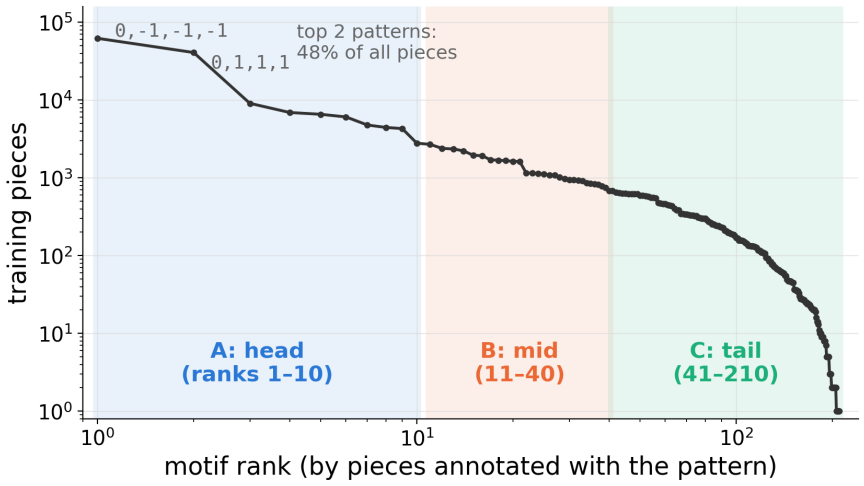


Figure 5: Motif rank frequency over the full training corpus (log log), ranked by how many pieces carry each pattern as their extracted motif annotation. Shaded bands mark the head, mid, and tail frequency regimes used for evaluation sampling.

NeurIPS Paper Checklist

1. **Claims**

Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

Answer: [Yes]

Justification: The abstract and introduction clearly state the main claims and contributions of the paper, accurately reflecting the theoretical and experimental results presented.

Guidelines:

- The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [Yes]

Justification: The paper mentions where limitations are met and how they are overcome throughout the paper.

Guidelines:

- The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: The paper contains neither theoretical claims nor results.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: All information and artifacts needed to reproduce the results are provided.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

Justification: Both the data and foundational model used are publicly known and available. Code is provided in Appendix A.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

Answer: [Yes]

Justification: All details are provided in Appendix A.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: The experimental results provided include confidence intervals needed to claim statistical significance.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: All resources needed are detailed in Appendix A.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research in the paper conforms with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The paper discusses and demonstrates the importance and viability of models built to take musical inputs intuitive to composers.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All data and models used are properly cited and credited.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The code and demo website is provided in Appendix A and the Abstract respectively.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper involves no crowdsourcing or human-subject experiments. All evaluation is automatic, and a human listening study is explicitly deferred to future work (Section 4).

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: No research with human subjects was conducted.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.

- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: The paper does not involve LLMs as any important, original, or non-standard component.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.