FROM SHAPE TO MUSIC: CONTOUR-CONDITIONED SYMBOLIC MUSIC GENERATION

Qiaoxi Zhang
Centre for Digital Music,
Queen Mary University of London
qiaoxi.zhang@qmul.ac.uk

Mathieu Barthet
Centre for Digital Music,
Queen Mary University of London
Aix-Marseille Univ CNRS PRISM
m.barthet@qmul.ac.uk

Anna Xambó Sedó
Centre for Digital Music,
Queen Mary University of London
a.xambosedo@qmul.ac.uk

ABSTRACT

We present a novel approach to symbolic music generation that enables users to guide melodic trajectories through abstract contour inputs. This work explores how intuitive, high-level melodic controls can influence the expressive shape of generated music, paving the way for tools that grant users more creative control. Our framework consists of two main stages. In the first stage, we extract melodic contour from symbolic music using the Ramer–Douglas–Peucker (RDP) algorithm, which simplifies the melodic trajectory while preserving its essential directional shape. Second, a Transformer-based generative model is conditioned on user-specified contour tokens to produce polyphonic continuations that follow the desired melodic shape. To strengthen contour adherence, we introduce a contour-aware loss based on cosine similarity between the target contour vector and the generated melodic motion.

Objective evaluations on the MAESTRO dataset indicate that the proposed model closely aligns with target contours while preserving overall musical quality comparable to a baseline Music Transformer. These results highlight the feasibility of high-level contour-conditioned music generation and point toward future applications that integrate contour-based inputs with modalities such as hand gestures in VR, enabling intuitive, real-time human–AI co-composition.

1. INTRODUCTION

Controllability in AI music generation is rapidly advancing, with notable progress in recent years, yet it remains an open and actively explored research area. Recent advances in controllable music generation have explored a wide range of conditioning inputs, including chord [1, 2, 3], rhythm [1, 2], style [4, 5], emotion [6], and even textual or visual modalities [7, 8, 9]. These approaches enable users to steer the generation process toward desired musical attributes. While a few prior works have considered contour-based conditioning, direct and flexible control over melodic structure remains relatively underexplored. Melodic contour captures the explicit shape and direction of pitch movement over time and is central to human perception and music composition [10].

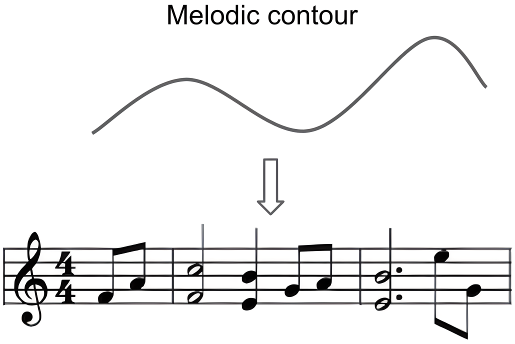

Figure 1. Illustration of abstract contour-conditioned music generation (Melody fragment inspired by Call of Silence, adapted by the authors).

Traditional instrumental performance often requires complex motor skills that are learned. Our longer-term design goal is to propose an alternative approach to music performance and composition that establishes more intuitive links between user actions and the resulting music, thereby lowering the learning barrier and enabling more direct expression of musical ideas. As a first step, we design a contour-conditioned music generation system and evaluate it quantitatively.

This study explores new ways for composers to shape machine-generated music through high-level melodic sketches. First, we present a novel method for extracting high-level melodic contours by combining time-pitch segmentation with Ramer–Douglas–Peucker (RDP) simplification, producing a compact yet expressive representation of melodic direction. We extend the Music Transformer [11] with contour tokens and a contour-aware loss, enabling the generation of polyphonic, musically coherent continuations aligned with user-defined contours and facilitating collaborative human–AI composition.

2. RELATED WORK

Melodic contour has been extensively studied in the context of audio-based music analysis, where the goal is to estimate and interpret pitch trajectories from audio signals [12, 13].

These approaches rely on fundamental frequency (F0) extraction and are often used in tasks such as music transcription, genre classification, or melody identification. In contrast, our work operates directly on symbolic music (e.g., MIDI), where pitch and onset information is precisely available. This enables a more direct and interpretable extraction of high-level contour, which we further use as a conditioning signal in generation.

For polyphonic symbolic music, a prerequisite for contour analysis is melody extraction, which aims to identify the main melodic line from multiple voices. Early rule-based methods such as the skyline algorithm [14] simply select the highest note at each onset. Later approaches evolved toward data-driven models: piano-roll based convolutional networks treat scores as images [15], while sequence-based recurrent neural networks capture contextual dependencies among notes [16]. Feature-based systems [17] use handcrafted note properties such as pitch, duration, and note dissonance. Recently, transformer-based pre-trained models like MIDI-BERT [18] have demonstrated strong performance on melody extraction as a downstream symbolic task, reflecting a shift from heuristic to representation learning approaches.

Prior research in symbolic music generation has extensively explored melody-conditioned approaches over the past decades. MusicGen [19] and MusicLM [7] transfer the style of an input melody based on textual descriptions or extracted melodic features. Works such as [20] and [21] investigate note-level contour manipulation: [20] proposes an approach to generate variations or mixtures of multiple contours, while Piano Genie [21] maps keystrokes from a simplified 8-key interface to a full piano keyboard, following the direction of pitch movement on the smaller keyboard. GaMaDHaNi [22] introduces expressive elements to music by modeling singers' vocal melodies extracted from audio recordings and generating ornamental notes for a given sequence. Music SketchNet [23] performs pitch interpolation by transforming a set of user-specified pitches into a complete sequence.

Some systems have also explored mapping abstract contours to melodies, which is more closely related to our work. For example, Hyperscore [24] introduced a melody-pattern-based composition interface, where different colours represent predefined melodic patterns and the drawn shapes control their transformations. This approach allows novice users to compose music by drawing, but its flexibility is constrained by the small set of predefined patterns available.

Pizzicato [25] allows users to draw a pitch curve to fill a measure. The system is rule-based, requiring explicit rhythm and chord labels for each beat. Notes are chosen as the closest pitch within the chord, independent of surrounding measures, which makes it unsuitable for users without prior music theory knowledge.

Another related system addresses melody inpainting by allowing users to draw curves representing the desired pitch contour, which the system then converts into a melody matching both rhythm and pitch contour [26]. Built on a variational auto-encoder with a melody disentanglement scheme, it enables interactive generation without requiring users to understand music notation. However, this system is limited to monophonic melodies and focuses on filling missing measures rather than generating full polyphonic music from abstract contours. In contrast, our work conditions a transformer-based model on high-level contour tokens to support polyphonic generation, allowing contour-driven composition beyond inpainting tasks.

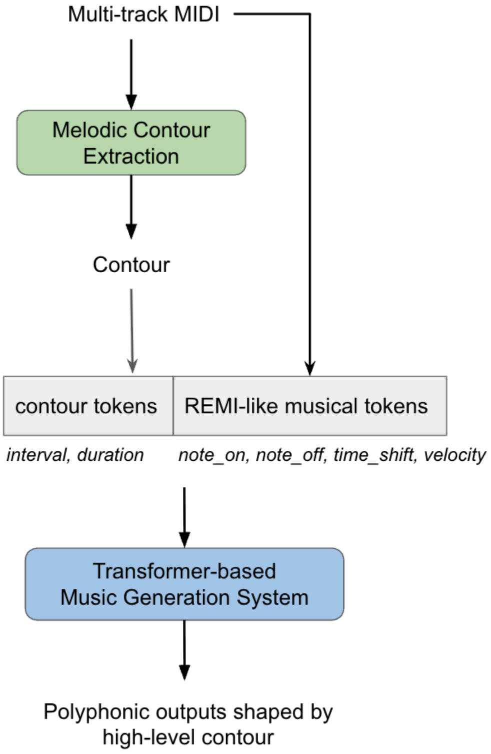
Multi-track MIDI
Melodic Contour Extraction
Contour
contour tokens
REMI-like musical tokens
interval, duration
note_on, note_off, time_shift, velocity
Transformer-based Music Generation System
Polyphonic outputs shaped by high-level contour

Figure 2. Overall pipeline of contour-conditioned symbolic music generation.

The system Drawlody [27] enables users to create melodies by sketching pitch contours, utilising a simplified contour representation and a CNN-Transformer architecture to map sketches into melody. Drawlody improves usability by eliminating the need for auxiliary musical inputs like chords or pre-set rhythms, making it accessible to non-expert users. However, it is limited to monophonic melody generation and does not support conditioning on higher-level polyphonic or structural context.

3. METHOD

We develop a contour-conditioned music generation framework that integrates high-level structural information into symbolic music modelling. The overall process consists of two main components: (1) contour extraction, which preprocesses symbolic MIDI files by simplifying melodic motion using an RDP-based algorithm, and (2) a Transformer-based model that learns to generate polyphonic music conditioned on the extracted contour. The contour is encoded as high-level event tokens and combined with REMI-like musical tokens to form the model input. These contour tokens serve as structural guidance during both training and generation. An overview of this pipeline is illustrated in Fig. 2.

3.1 Contour Extraction using RDP

The dataset used in this project is MAESTRO v2.0.0, a polyphonic piano MIDI corpus [28]. Our contour extraction process involves first deriving a simplified monophonic melody line from the original polyphonic music, followed by encoding the contour information in a format specifically designed to support model training.

The first step is to extract the main melody line, i.e., to identify a monophonic sequence from the polyphonic texture. As a proof of concept, we employed a simplified strategy that selects notes based on pitch and velocity. Specifically, within each sliding window of 1 second (with a hop size of 0.5 seconds), the note with the highest pitch is chosen, with velocity serving as a secondary criterion to break ties or slightly favour louder notes. Here, velocity is weighted roughly one-tenth of the pitch value in the selection process. The pitch–time trajectory of the selected notes is then recorded. Fig. 3 illustrates the result of extracting a monophonic melody line from polyphonic input.

The next and most crucial step is melody simplification, where we apply the Ramer–Douglas–Peucker (RDP) algorithm to identify key turning points in the melodic trajectory while discarding less structurally relevant notes [29]. Originally developed for applications such as computer graphics and cartography, RDP is widely used to simplify complex curves and reduce data while preserving essential shape characteristics. The RDP algorithm recursively eliminates points whose perpendicular distance to the line formed by the first and last points in a segment is below a predefined threshold. In our adaptation, we apply the RDP algorithm recursively using a dynamic threshold tailored to the pitch range of each piece. The process begins with a threshold equal to the smallest pitch difference between adjacent notes and gradually increases, eliminating points until the number of remaining notes falls below a fixed ratio of the total notes in the monophonic sequence. We empirically tuned this ratio to 1/3 to simplify the contour while preserving melodic shape granularity; however, optimisation techniques could be applied in future work to determine a more optimal value.

Finally, we extract melodic contour information from the simplified melody. Each contour segment is represented by a pair: interval and duration, corresponding respectively to the pitch and time differences between the first and last notes of the segment. To produce a more abstract representation of melodic direction, we apply a merging step: consecutive segments with the same direction (ascending, descending, or flat) are combined, and segments with a shorter duration are merged with adjacent segments. Empirically, segments shorter than 2 seconds are merged as they typically reflect fine-grained melodic variation rather than the overall contour. Fig. 4 provides an overview of the full melodic contour extraction pipeline, illustrating the intermediate steps and final result.

3.2 Model Architecture

We base our architecture on the Music Transformer [11], a well-known autoregressive model with relative positional encoding (RPR). Rather than modifying the core architecture of the Music Transformer, we extend its input token vocabulary by introducing two additional event types to represent melodic contour information (see Section 3.3). These contour tokens are injected into the input sequence and processed identically to other events, allowing the model to condition on high-level melodic direction without altering its internal structure.

We enable Relative Positional Representation (RPR) throughout training and inference, which improves the model's ability to capture positional relations between events over long spans. Additionally, we apply label smoothing with $\varepsilon = 0.1$ in the cross-entropy loss to reduce overconfidence in token prediction and enhance generalisation, especially for autoregressive generation tasks with large output vocabularies.

3.3 Contour Token Design

The data is represented using a REMI-like event-based vocabulary [30] consisting of note_on, note_off, time_shift, and velocity tokens. To incorporate high-level melodic guidance, we introduce two additional token types: contour_interval and contour_duration, which respectively represent the expected pitch movement and temporal span of a melodic segment. These contour tokens are inserted immediately before the group of note events they describe. An example token sequence is shown below:

```
<Event type: contour_interval, value: 85>
<Event type: contour_duration, value: 160>
<Event type: time_shift, value: 99>
<Event type: time_shift, value: 0>
<Event type: velocity, value: 10>
<Event type: note_on, value: 62>
<Event type: time_shift, value: 1>
<Event type: velocity, value: 10>
```

Based on an empirical analysis of the dataset, we define the unit and range for each contour token type. contour_interval is measured in semitones, with 201 discrete values ranging from -100 to $+100$ (inclusive), and contour_duration is measured in deciseconds (0.1 seconds), using 150 discrete values from 0.1 to 15.0 seconds. As a result, the overall vocabulary is expanded by 351 tokens to accommodate the additional contour information.

3.4 Contour-Aware Loss

To guide the model to follow the desired melodic contour, we introduce a contour-aware loss that aligns the movement of generated notes with the contour direction. Each contour block is defined by a pair of tokens: `<contour_interval>` and `<contour_duration>`, which precede a group of notes in the token sequence. The interval represents the expected pitch change, and the duration defines the expected temporal span.

We extract vectors representing melodic motion within each contour block by uniformly dividing the interval

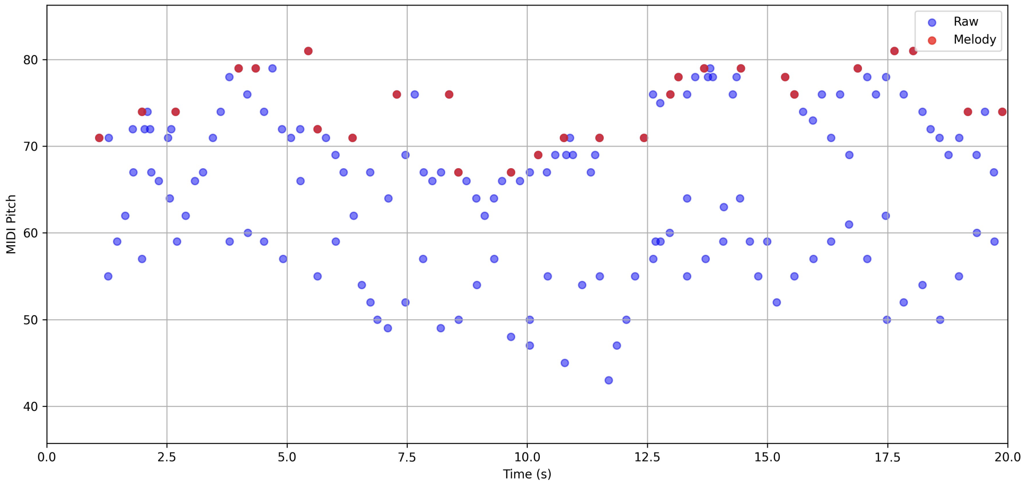

Figure 3. Example of monophonic melody extraction from polyphonic symbolic input. The extracted melody (red) captures the main pitch trajectory selected from the full polyphonic texture (purple).

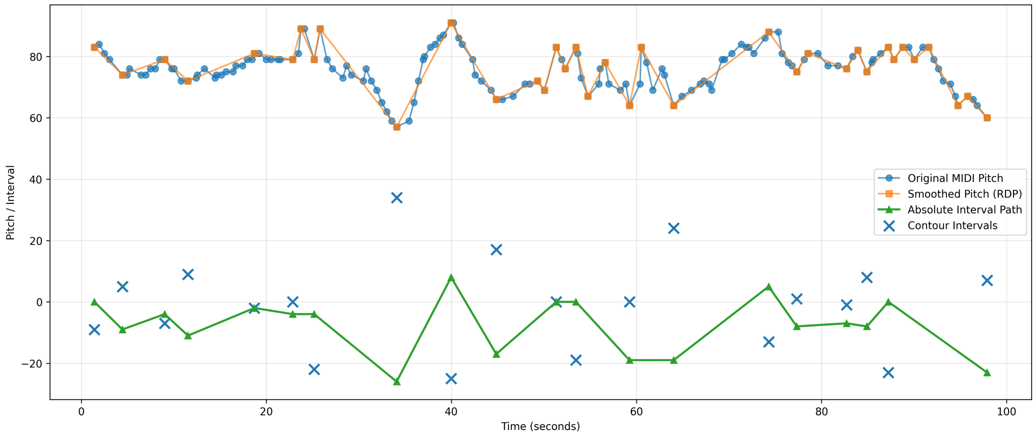

Figure 4. Visualisation of the complete melodic contour extraction pipeline. The figure shows the original MIDI pitch sequence, the RDP-simplified melody, the absolute interval path, and the final extracted contour intervals. The value of each contour interval (blue crosses) indicates the pitch difference between the current and the next blue cross. The absolute interval path is reconstructed from the contour intervals to facilitate comparison with the shape of the original MIDI sequence.

(e.g., into 5 equal-length time slices), and selecting the highest-pitched note in each slice. Each note vector encodes its relative time and pitch difference from the first note in the block. A contour vector is similarly defined as $(\Delta t, \Delta p)$, where Δt is the total contour duration and Δp is the contour interval.

We define the cosine margin loss between note vectors $\mathbf{n}_i$ and the contour vector $\mathbf{c}$ as given in (1).

$$\mathcal{L}_{\text{contour}} = \frac{1}{N}\sum_{i=1}^{N} \max\left(0, m - \cos(\theta_i)\right) \tag{1}$$

where θ_i is the angle between $\mathbf{n}_i$ and $\mathbf{c}$, and m is a fixed cosine similarity margin (empirically set to 0.87, corresponding to an angle of approximately 30 degrees between $\mathbf{n}_i$ and $\mathbf{c}$). This encourages note movements to align with the direction of the contour vector, while allowing flexibility for natural musical variation.

To balance the contour-aware loss and the token-level cross-entropy loss, we introduce a weighting factor λ_{contour} that gradually increases during training. This allows the model to first focus on learning basic token structure before contour-level alignment. The λ scheduler is defined as given in (2).

$$\lambda_{\text{contour}}(e) = \begin{cases} 0, & \text{if } e < 15 \\ 0.02 \cdot (e - 15), & \text{if } 15 \leq e < 30 \\ 0.3, & \text{if } e \geq 30 \end{cases} \tag{2}$$

where e denotes the current epoch. The final training objective is the weighted sum of cross-entropy loss $\mathcal{L}_{\text{CE}}$ and contour-aware loss $\mathcal{L}_{\text{contour}}$, as shown in (3).

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{CE}} + \lambda_{\text{contour}} \cdot \mathcal{L}_{\text{contour}} \tag{3}$$

4. EXPERIMENTS AND RESULTS

4.1 Experiment Setup

4.1.1 Dataset and Data Preprocessing

We conduct our experiments on the MAESTRO v2.0.0 dataset [28], a large-scale collection of virtuosic piano performances comprising approximately 200 hours of paired MIDI and audio recordings. In this work, we use only the MIDI portion of the dataset, which provides precise symbolic representations of note events (pitch, velocity, timing) necessary for contour-level modeling and event-based tokenisation. The dataset features expressive, polyphonic piano music with wide pitch coverage across registers, providing a rich and versatile context for learning melodic contours and supporting contour-aware generation. Furthermore, MAESTRO was also used to train the original Music Transformer [11], ensuring compatibility and reproducibility in our architecture and training pipeline.

Each MIDI file is tokenised using a REMI-like vocabulary consisting of note, velocity, and time-shift events. We introduce two new event types: `<contour_interval>` and `<contour_duration>`, which represent contour direction and span as described in Section 3. Time shifts are quantised to 10ms, and contour durations use a coarser unit of 100ms.

4.1.2 Model Configuration

We base our model on the Music Transformer with Relative Positional Representations (RPR) [11]. The model dimension is set to $d_{\text{model}} = 512$ with 8 attention heads, consistent with the original configuration. To ensure sufficient model capacity for learning extended contour-aware sequences, we added two Transformer layers compared with the original model. The maximum sequence length is set to 768 tokens, sufficient to cover a contour span of up to 15 seconds, whereas the original Music Transformer used sequences of up to 2048 tokens to model entire pieces. The model is trained with a batch size of 16 and label smoothing ($\varepsilon = 0.1$).

4.1.3 Training Details

We train the model for up to 300 epochs using the Adam optimizer. To encourage progressive learning of contour structure, we employ a curriculum-like scheduling for the contour-aware loss (Section 3), where the weight λ_{contour} is linearly increased over the first 15 epochs and held constant afterward. This allows the model to first learn local structure before aligning to high-level contour.

4.2 Contour Extraction Analysis

We assess our contour representation on MAESTRO v2.0.0 files, comparing the extracted contour against the monophonic melody line obtained in preprocessing. The method achieves a mean compression ratio of **0.1362** ± **0.0141**, retaining only ∼14% of the original notes while capturing overall melodic direction. Root Mean Square Error (RMSE), widely used to quantify reconstruction accuracy in symbolic or audio-based music generation [31, 32], measures the average deviation between predicted and reference values. Pitch fidelity is measured via RMSE between reconstructed and original trajectories, yielding **8.20** ± **2.44** semitones. Despite the strong abstraction, visual comparisons (Fig. 4) show that the contour preserves key turning points and the global rise-and-fall pattern of the melody. These results suggest that our RDP-based approach produces a compact yet musically meaningful representation suitable for guiding high-level controllable music generation.

4.3 Music Generation Objective Evaluation

We evaluate the generated music using two main approaches: musicality assessment via MusPy metrics [33] and contour alignment analysis.

4.3.1 Musicality Metrics

We compute several symbolic music descriptors using MusPy [33], as shown in Table 1. The *number of pitch classes used* and *pitch entropy* measure melodic diversity, while the *polyphony rate* captures the complexity of polyphonic textures. The *empty measure rate* and *groove consistency* describe rhythmic completeness and stability.

Results show that the proposed contour-conditioned model (MT+CC) performs similarly to the baseline Music Transformer (MT) model across most musicality metrics, indicating that contour conditioning does not negatively impact overall musicality. MT achieves empty measure rates closer to the groundtruth, indicating better alignment with human-like phrasing. In contrast, MT+CC produces fewer empty measures, likely because specifying a target contour forces the model to generate notes throughout the conditioned segment, leaving less room for natural pauses or silence.

4.3.2 Contour Alignment Metrics

Given a primer clip and a target contour defined by an interval and a duration (see Fig. 5), the model generates a polyphonic continuation. For evaluation, we extract a monophonic melody using a window-based selection method and fit a linear regression to the note pitches over time. We then compute the **slope ratio**, defined as the generated slope divided by the target slope, which measures how well the melodic trajectory aligns with the intended contour. A slope ratio close to 1 indicates strong alignment with the target contour. A slope ratio greater than zero indicates that the generated melody follows the same directional trend as the target contour, whereas a negative value indicates an opposite direction. The absolute value reflects the relative steepness: values larger than one indicate faster pitch changes (steeper melodic motion), while values smaller than one suggest a gentler slope compared with the target contour.

As shown in our results, MT+CC achieves a slope ratio of 1.20 on average, indicating that contour conditioning successfully biases the melodic direction towards the target contour. The ablation model, without contour conditioning, achieves a negative slope ratio (-0.17), demonstrating a lack of alignment with the target direction.

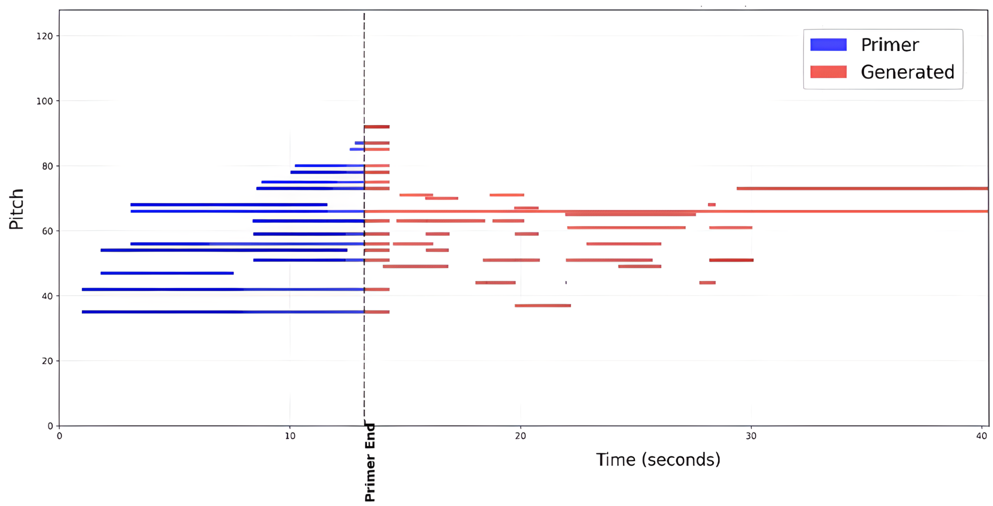

Figure 5. Contour-conditioned generation example. Given the condition to descend by 30 semitones over 10 seconds, the generated melody follows the target contour, dropping from approximately pitch 90 to 60 between 13–23 seconds.

Model	pitch classes	pitch entropy	polyphony rate	empty measure rate	groove consistency
MAESTRO v2.0.0	11.98	5.3762	0.3305	0.0081	0.9762
Music Transformer (MT)	10.26	4.3819	0.6782	0.0075	0.9841
MT+contour-conditioning	10.36	4.4654	0.6591	0.0056	0.9836

Table 1. Comparison of mean musicality metrics between the groundtruth dataset (MAESTRO v2.0.0), the baseline Music Transformer (MT), and the proposed contour-conditioned model (MT+CC).

4.3.3 Interactive Generation

To demonstrate potential user-facing applications of contour-conditioned generation, we implemented a simple interface where users can specify a sequence of contour conditions over time (a rolling window). Unlike real-time systems, the current prototype processes the entire input sequence offline and produces a complete continuation before playback.

Generated examples are available on Zenodo. [1]

5. DISCUSSION

The current simplified strategy for extracting a monophonic melody from polyphonic symbolic music can approximate a main melodic line, but errors and ambiguities are inevitable. Misalignments in timing or selecting harmonically relevant but non-melodic notes can distort the intended melodic contour. As a result, the evaluation of contour-conditioned music generation systems suffers from noise in the extracted reference melody, making it difficult to reliably quantify alignment between generated music and target contours.

Despite these limitations, contour-conditioned generation has strong potential in real-world applications. By allowing users to specify only a high-level contour—through drawing a line or humming a rough pitch trajectory—such systems lower the barrier to music creation. Non-professional users can express musical intent without providing precise pitches, rhythms, or chords, fostering accessibility and democratisation of composition tools. In educational contexts, contour-based interfaces could help students visualise melodic motion and explore variations interactively. Similarly, they could enhance live improvisation setups, where performers guide generative systems via intuitive gestures.

Looking ahead, contour conditioning enhances the controllability and intuitiveness of music AI. This paradigm may serve as a foundation for future interactive systems that combine computational generation with human compositional intent.

6. CONCLUSIONS

This work presented an initial exploration of abstract contour-conditioned symbolic music generation. We introduced a method to extract melodic contours via time-pitch segmentation and RDP simplification, and integrate these contour tokens into a Music Transformer model. Our experiments demonstrated the feasibility of conditioning symbolic music generation on abstract melodic contours, although the results were limited by current melody extraction heuristics and the lack of robust objective evaluation tools.

Future research should focus on improving melody extraction techniques, for example by leveraging data-driven methods to isolate a more musically meaningful main melody line, which would also benefit evaluation reliability. Expanding the conditioning scheme to incorporate additional musical context, such as backing tracks or harmonic information, could increase controllability and improve musical coherence. Human

[1] DOI: `https://doi.org/10.5281/zenodo.17322509`

evaluations are needed to complement automated metrics and better capture perceptual alignment with contour intent. Finally, we envision applications where contour-based input is combined with novel modalities, such as hand gesture control in VR environments, enabling intuitive and expressive human–AI co-composition with real-time responsiveness, where latency and interaction design remain open challenges.

7. ACKNOWLEDGMENT

This work is supported by the UKRI and EPSRC under grant EP/S022694/1.

8. REFERENCES

[1] Y.-H. Lan, W.-Y. Hsiao, H.-C. Cheng, and Y.-H. Yang, "MusiConGen: Rhythm and chord control for transformer-based text-to-music generation," in *Proceedings of the 25th International Society for Music Information Retrieval Conference*. ISMIR, Nov. 2024, pp. 311–318.

[2] O. Tal, A. Ziv, I. Gat, F. Kreuk, and Y. Adi, "Joint audio and symbolic conditioning for temporally controlled text-to-music generation," in *Proceedings of the 25th International Society for Music Information Retrieval Conference*. ISMIR, Nov. 2024, pp. 264–271.

[3] Z. Guo and S. Dixon, "Moonbeam: A MIDI foundation model using both absolute and relative music attributes," *arXiv preprint arXiv:2505.15559*, 2025.

[4] G. Brunner, Y. Wang, R. Wattenhofer, and S. Zhao, "Symbolic music genre transfer with CycleGAN," in *2018 IEEE 30th International Conference on Tools with Artificial Intelligence (ICTAI)*. IEEE, 2018, pp. 786–793.

[5] K. Choi, C. Hawthorne, I. Simon, M. Dinculescu, and J. Engel, "Encoding musical style with transformer autoencoders," in *International Conference on Machine Learning*. PMLR, 2020, pp. 1899–1908.

[6] H.-T. Hung, J. Ching, S. Doh, N. Kim, J. Nam, and Y.-H. Yang, "EMOPIA: A multi-modal pop piano dataset for emotion recognition and emotion-based music generation," in *Proceedings of the 22th International Society for Music Information Retrieval Conference*, 2021, pp. 318–325.

[7] A. Agostinelli, T. I. Denk, Z. Borsos, J. Engel, M. Verzetti, A. Caillon, Q. Huang, A. Jansen, A. Roberts, M. Tagliasacchi *et al.*, "MusicLM: Generating music from text," *arXiv preprint arXiv:2301.11325*, 2023.

[8] H. Liu, Z. Chen, Y. Yuan, X. Mei, X. Liu, D. Mandic, W. Wang, and M. D. Plumbley, "AudioLDM: Text-to-audio generation with latent diffusion models," in *Proceedings of the 40th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett, Eds., vol. 202. PMLR, 23–29 Jul 2023, pp. 21 450–21 474.

[9] Z. Wang, C. Bao, L. Zhuo, J. Han, Y. Yue, Y. Tang, V. S.-J. Huang, and Y. Liao, "Vision-to-music generation: A survey," *arXiv preprint arXiv:2503.21254*, 2025.

[10] W. J. Dowling, A. Barbey, and L. Adams, "Melodic and rhythmic contour in perception and memory," *Music, mind, and science*, pp. 166–188, 1999.

[11] C. A. Huang, A. Vaswani, J. Uszkoreit, N. Shazeer, C. Hawthorne, A. M. Dai, M. D. Hoffman, and D. Eck, "An improved relative self-attention mechanism for transformer with application to music generation," *CoRR*, vol. abs/1809.04281, 2018.

[12] A. De Cheveigné and H. Kawahara, "YIN, a fundamental frequency estimator for speech and music," *The Journal of the Acoustical Society of America*, vol. 111, no. 4, pp. 1917–1930, 2002.

[13] J. W. Kim, J. Salamon, P. Li, and J. P. Bello, "CREPE: A convolutional representation for pitch estimation," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 161–165.

[14] A. Uitdenbogerd and J. Zobel, "Melodic matching techniques for large music databases," in *Proceedings of the Seventh ACM International Conference on Multimedia (Part 1)*, 1999, pp. 57–66.

[15] F. Simonetta, C. E. Cancino-Chacón, S. Ntalampiras, and G. Widmer, "A convolutional approach to melody line identification in symbolic scores," in *Proceedings of the 20th International Society for Music Information Retrieval Conference (ISMIR)*, 2019.

[16] Y.-W. Hsiao and L. Su, "Learning note-to-note affinity for voice segregation and melody line identification of symbolic music data," in *Proceedings of the 22nd International Society for Music Information Retrieval Conference (ISMIR)*, 2021, pp. 285–292.

[17] H. Zhao and Z. Qin, "TuneRank model for main melody extraction from multi-part musical scores," in *2014 Sixth International Conference on Intelligent Human-Machine Systems and Cybernetics*, vol. 2. IEEE, 2014, pp. 176–180.

[18] Y.-H. Chou, I.-C. Chen, C.-J. Chang, J. Ching, and Y.-H. Yang, "Bert-like pre-training for symbolic piano music classification tasks," *Journal of Creative Music Systems*, vol. 8, no. 1, pp. 1–19, 2024.

[19] J. Copet, F. Kreuk, I. Gat, T. Remez, D. Kant, G. Synnaeve, Y. Adi, and A. Défossez, "Simple and controllable music generation," *Advances in Neural Information Processing Systems*, vol. 36, pp. 47 704–47 720, 2023.

[20] T. Akama, "Controlling symbolic music generation based on concept learning from domain knowledge." in *Proceedings of the 20th International Society for Music Information Retrieval Conference*, 2019, pp. 816–823.

[21] C. Donahue, I. Simon, and S. Dieleman, "Piano Genie," in *Proceedings of the 24th International Conference on Intelligent User Interfaces*, 2019, pp. 160–164.

[22] N. N. Shikarpur, K. M. Dendukuri, Y. Wu, A. Caillon, and C.-Z. A. Huang, "Hierarchical generative modeling of melodic vocal contours in hindustani classical music," in *Proceedings of the 25th International Society for Music Information Retrieval Conference*. ISMIR, Nov. 2024, pp. 1020–1028.

[23] K. Chen, C.-i. Wang, T. Berg-Kirkpatrick, and S. Dubnov, "Music SketchNet: Controllable music generation via factorized representations of pitch and rhythm," in *Proceedings of the 21th International Society for Music Information Retrieval Conference*, 2020, pp. 77–84.

[24] M. M. Farbood, E. Pasztor, and K. Jennings, "Hyperscore: a graphical sketchpad for novice composers," *IEEE Computer Graphics and Applications*, vol. 24, no. 1, pp. 50–54, 2004.

[25] Arpege-Music, "Pizzicato notation software," http://www.arpegemusic.com/manual36/EN855.htm, 2025, online; accessed 20 July 2025.

[26] C. Benetatos and Z. Duan, "Draw and listen! a sketch-based system for music inpainting," *Transactions of the International Society for Music Information Retrieval*, vol. 5, no. 1, 2022.

[27] Q. Liang and Y. Wang, "Drawlody: Sketch-based melody creation with enhanced usability and interpretability," *IEEE Transactions on Multimedia*, vol. 26, pp. 7074–7088, 2024.

[28] C. Hawthorne, A. Stasyuk, A. Roberts, I. Simon, C.-Z. A. Huang, S. Dieleman, E. Elsen, J. Engel, and D. Eck, "Enabling factorized piano music modeling and generation with the MAESTRO dataset," in *International Conference on Learning Representations*, 2019.

[29] D. H. Douglas and T. K. Peucker, "Algorithms for the reduction of the number of points required to represent a digitized line or its caricature," *Cartographica: the International Journal for Geographic Information and Geovisualization*, vol. 10, no. 2, pp. 112–122, 1973.

[30] 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.

[31] A. Kasif, S. Sevgen, A. Ozcan, and C. Catal, "Hierarchical multi-head attention lstm for polyphonic symbolic melody generation," *Multimedia Tools and Applications*, vol. 83, no. 10, pp. 30 297–30 317, 2024.

[32] J. Liu, C. Li, Y. Ren, Z. Zhu, and Z. Zhao, "Learning the beauty in songs: Neural singing voice beautifier," *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, p. 7970–7983, 2022.

[33] H.-W. Dong, K. Chen, J. McAuley, and T. Berg-Kirkpatrick, "MusPy: A toolkit for symbolic music generation," in *Proceedings of the 21th International Society for Music Information Retrieval Conference*, 2020, pp. 101–108.