Drawlody: Sketch-Based Melody Creation with Enhanced Usability and Interpretability

Qihao Liang, Ye Wang *Member, IEEE*

***Abstract*—Sketch-based melody creation systems enable people to easily compose melodies by converting human-sketched melody contours into coherent melodies that fit the depicted contours. While this remains one of the most intuitive approaches to interactive music creation, previous studies are still stagnating in some usability and interpretability issues. For one thing, these studies entail additional complex musical conditions as auxiliary inputs (e.g. chord progressions, contextual melodies, and predetermined rhythms), supporting only fixed-length and rule-based melody generation. Moreover, users without enough musical expertise might find it difficult to define appropriate inputs to these systems, as they cannot interpret the role of their inputs in guiding melody generation. To address these limitations, we present Drawlody, a sketch-based melody creation system with enhanced usability and interpretability. Specifically, Drawlody simplifies user input requirements by excluding all complex musical conditions, using only a simplified melody contour representation named Generalised Melody Contour (GMC) as input. This simplification clarifies the role of user controls, making the system more usable for people without musical training. To guide coherent melody generation from GMC, we propose FlexMIDI music representation, which simulates the tonal structure of melodies and faithfully explains how human-sketched contours guide melody generation. We employ a CNN-Transformer-based architecture as the foundation model to achieve arbitrary-length melody generation. Drawlody is evaluated by both objective and subjective music quality studies, as well as a usability and interpretability study. The results support its enhanced usability, interpretability, and high-quality melody generation capabilities. Video demonstrations of the system are presented here.**

***Index Terms*—Music Generation; Interactive Music Creation; Melody Contour**

I. Introduction

RECENT advances in artificial intelligence have significantly progressed machine music composition through the prowess of generative models, including but not limited to Recurrent Neural Networks (RNNs) [22], [34], [38], [50], Convolutional Neural Networks (CNNs) [8], and different variants of attention-based models [28], [29], [31], [53], [57], [60], [71]. To enhance the usability and controllability of these complex architectures, researchers have attempted to develop some interfaces [3], [13], [15], [25], [55], [56], [59], [67] that allow users to interact with generative models and guide them in composing music that aligns better with users' expectations.

Qihao Liang and Ye Wang are affiliated with the School of Computing, National University of Singapore, Singapore. Ye Wang is the correspondence author of this paper. This paper has supplementary downloadable material available at http://ieeexplore.ieee.org., provided by the author. The material includes some technical details of algorithms and example demonstrations used in this study. This material is 2.7MB in size.

Manuscript received 9th September, 2023; revised 30th December, 2023.

Nonetheless, previous interactive music creation (IMC) systems still stagnate in a trade-off between usability and interpretability. Some of them prioritise user experience with intuitive interfaces, such as emotion-based [12], [32], [72] and picture style-based music generation frameworks [52]. While these systems provide user-friendly interactions using easily obtainable media as input, the effects of user controls are often ambiguous, as generative models do not provide human-understandable explanations of how these input controls guide music generation. For instance, when using emotions or pictures as input controls, users may find the generated music less relevant or even contrary to their expectations, nor can they interpret how the generated music relates to their input. This is largely because the perception of emotions and picture styles is highly subjective and incompletely formulated by domain knowledge [26]. In contrast, other IMC studies emphasise the integration of clearly defined domain knowledge to make the AI music generation process more interpretable to humans, enhancing the "domain expertise" of generative models. Examples include but are not limited to melody generation from chords [10], [14], the use of unfinished human compositions to guide music generation [3], [45], [46], [62], and Digital Audio Workstations (DAWs) that leverage digital signal theories to create specific sound effects [11]. However, the excessive reliance on domain knowledge of these systems can result in reduced usability, especially for amateur users without sufficient expertise in music.

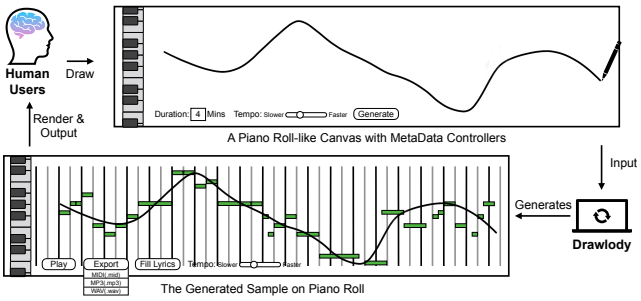

Fig. 1. The general idea of Drawlody. Users can sketch a simple curve on the canvas, without having to provide other musical conditions (e.g. chords, contexts, etc.). The system takes this sketch and outputs a musically coherent melody whose pitch motion also generally matches this curve.

To address these issues, we propose Drawlody—an interactive music creation framework with enhanced usability and interpretability. This framework is based on the notion of melody generation from melody contour sketches. Unlike previous sketch-based studies that (1) support only fixed-length rule-based music creation and (2) involve other complex musical conditions as input [5], [36], [37], Drawlody supports arbitrary-length melody generation from only melody contour

sketches (Figure 1). To cater to amateur users without a music background, Drawlody excludes all complex music theory-related input requirements (e.g. chords, prompt melodies) and indecipherable subjective attributes (e.g. emotions, picture styles, video styles), using only the simplified generalised melody contour (GMC) as user input. GMC omits the peripheral details of melody contours and retains only the fundamental pitch motion of melodies, enhancing the system usability by making the interaction more straightforward. Furthermore, GMC is also a faithful visual indicator of the general pitch motion of melodies, clarifying the role of user controls compared to using other ambiguous attributes (e.g. emotions or images) as input.

In addition to the simplification of user tasks, another challenge is to guide generative models in composing coherent melodies from users' sketches, and to enhance the system interpretability. Inspired by the notion of tonal music [58], we propose FlexMIDI melody representation, which represents a melody as its general trend (viz. the tonal centres) and finer details (viz. various pitches around tonal centres). This strategy ensures the alignment between input sketches and general trends of melodies, while allowing for the generation of more finer details to sustain melodic coherence and flexibility. The representation also explains how users' sketches influence and relate to the generated melodies, making the system more interpretable to users.

After modelling both sketches and melodies, we define Drawlody as an end-to-end cross-modal sequence generation task, employing a CNN-Transformer-based architecture as the foundation model to generate arbitrary-length monophonic melodies from melody contour sketch images. To evaluate Drawlody, we conducted a music quality study with objective and subjective metrics, as well as a usability and interpretability study. For the music quality study, we selected two sketch-based baselines: a rule-based genetic algorithm [36] and a CNN-based [37] model. To understand the effectiveness of our proposed GMC and FlexMIDI, we also implemented another three baselines under raw melody contours, MIDI and MuMIDI representations [49]. All baseline approaches were trained and validated on a synthetic dataset, and tested on a human-grounded dataset. Both objective and subjective evaluations advocate the advantage of Drawlody over other baselines in generating higher-quality melodies.

For the usability and interpretability study, we employed proxy-grounded and human-grounded tasks, two established benchmarks from the field of interpretable machine learning [20]. We chose five representative systems in the field of interactive music creation as baselines. The outcomes demonstrate significantly higher improvement in usability and interpretability of Drawlody against other IMC systems.

In summary, our main contributions include:

1. We present Drawlody, a sketch-based music creation framework with enhanced usability and interpretability.

2. We propose generalised melody contour (GMC), a novel concept that makes Drawlody accessible to amateur users and can also effectively control the overall trend of machine-generated melodies.

3. We propose FlexMIDI representation, helping generative models compose coherent melodies from GMC with clearer tonal structures and human-understandable explanations.

4. We leverage music domain knowledge to improve the quality of sketch-based melody generation, meanwhile assisting humans to interpret how their sketches control the generated melodies.

5. We explore the interpretability of Drawlody with some benchmarks from the field of interpretable machine learning.

II. Related Work

In the context of Human-Computer Interaction (HCI), the notion of interactive artificial intelligence has garnered significant attention in many recent studies [1], [63], [65], [66]. This trend has prompted computer scientists and musicians to explore the incorporation of HCI principles into automatic music generation frameworks. In this context, we present a taxonomy of previous studies on Interactive Music Creation (IMC) by categorising studies as usability-centred, interpretability-centred and sketch-based.

Usability-centred studies seek to develop interaction methods that are user-friendly and limit user workload. Such studies often utilise easily obtainable media as input to guide music generation, but often stop short of explaining *how* user input relates to the generated music. For example, prior work has explored the use of emotions as input for machine-generated music [2], [32], [33], [72], which often involves the analysis of facial expressions [73], body movements [12] and emotion models [19]. However, the computational analysis of emotions in both humans and music remains challenging due to the subjective and ambiguous nature of understanding emotions [35], [69]. This inherent complexity poses difficulties in creating emotion-based interactions that are easily interpretable to users. Another relevant example is music generation from video [47], [52], text [9], [48], or image style [17], [64], which also lacks interpretability due to opaque relationships between the provided input and resulting music.

Interpretability-centred studies, in contrast, seek to leverage clearly defined domain knowledge to make AI's music generation process more interpretable to humans and faithful to music theory. One typical example is melody-to-chord generation [25], [40] and chord-to-melody generation [34], [49], [54], where generative models are trained to capture the regularity of melody-chord relationships. In addition, some scholars have proposed using unfinished human compositions to guide AI music generation, with a strategy similar to prompt-based generation from natural language processing, which can be easily implemented with [24], [28], [30]. Some commercial products, such as digital audio workstations (DAW), leverage many interpretable digital processing algorithms [7], [18], [27] to process music for various sound effects. However, the use of excessive domain knowledge in these studies often overwhelms non-expert users, resulting in reduced usability.

To the best of our knowledge, the notion of **sketch-based music composition** has only appeared in a few recent studies. The first instance is a music improvisation system [36], [37], which allows users to improvise short melodies during the playback of a given accompaniment. However, [36] generates

rhythms and pitches independently and considers only the coherence between two pitches with bi-grams. The rhythms are also monotonous due to the rule-based nature. Although [37] further extends [36] with a CNN to improve melodic diversity, it is still limited to fixed-length melody generation, analogous to [42]. Furthermore, both [37] and [36] entail chord progressions as conditional inputs, which non-expert users cannot provide. Another relevant work is a VAE-based melody inpainting system [5], where users can fill a blank measure between two contextual measures by drawing a pitch curve and a note density curve. However, this system is limited to measure-long music composition and requires two context melodies that can be unobtainable to amateur users. Additionally, the use of note density curves might appear somewhat counterintuitive, since human perception of rhythm is often associated with movement-based concepts [4], [39], which cannot be adequately represented by wavy curves.

III. Problem Formulation

Different from previous systems that condition melody generation on sketches and other music-related constraints, our notion of sketch-based melody generation comes from the basic element analysis of music [21], [41], which decomposes a melody into pitches and rhythms. Drawlody focuses on controlling the pitch motion of output melodies with human-sketched curves, while allowing for the flexible generation of various rhythmic patterns learnt from training data.

Let $\mathcal{D} = \{(\mathbf{G}^{(i)}, \mathbf{M}^{(i)})_{i=1}^{C}\}$ denote the corpus with C paired sketch-melody samples, where each $\mathbf{G}^{(i)} \in \mathbb{R}^{w \times h}$ represents a $w \times h$ sketch image, and each $\mathbf{M}^{(i)} = \{m_0^{(i)}, m_1^{(i)}, ..., m_{t-1}^{(i)}\}$ is a sequence of melody symbols. Drawlody employs an end-to-end generative model D_θ parameterised by θ, which generates symbolic melodies from given sketch images and some user-specified metadata $\mathbf{U}^{(i)}$ (e.g. tempo, length of melody, etc.):

$$\mathbf{M}^{(i)} = D_\theta(\mathbf{G}^{(i)}, \mathbf{U}^{(i)}) \tag{1}$$

At the t-th timestep, D builds the conditional probability p_t:

$$p_t = P(m_t^{(i)} | m_{0:t-1}^{(i)};\ \mathbf{G}^{(i)};\ \mathbf{U}^{(i)}; \theta) \tag{2}$$

We aim to optimise θ by reducing the cross-entropy loss:

$$\hat{\theta} = \arg\min_{\theta} \mathcal{L}_{train} := -\frac{1}{t} \sum_{j=0}^{t-1} r_{ij} m_j^{(i)} \log(p_j) \tag{3}$$

where r_{ij} represents the loss weight for the symbol $m_j^{(i)}$.

IV. Method

In this section, we first present a brief overview of the Drawlody system. Then, we describe the motivation and details of representing melody contour sketches and melodies. Finally, we delve into the technicalities of Drawlody architecture and how the architecture leverages and interacts with these representations.

A. System Overview

As shown in Figure 2, Drawlody adopts an encoder-decoder architecture as the foundation model. The encoder (Figure 2(A)) first employs a sketch image splitter to process each input image as sequential data. This sequence then passes through an input processing module (Figure 2(B)), where a CNN Feature Extractor, a General Pitch Extractor and Positional Encoding are applied to enhance the feature representation. The outputs of these three components are concatenated and linearly projected as conditional inputs to a 4-layer Transformer-XL [16] encoder. The hidden states from the encoder are taken by another 4-layer Transformer-XL to sample musical symbols, which are finally transformed by an output module to a MIDI melody.

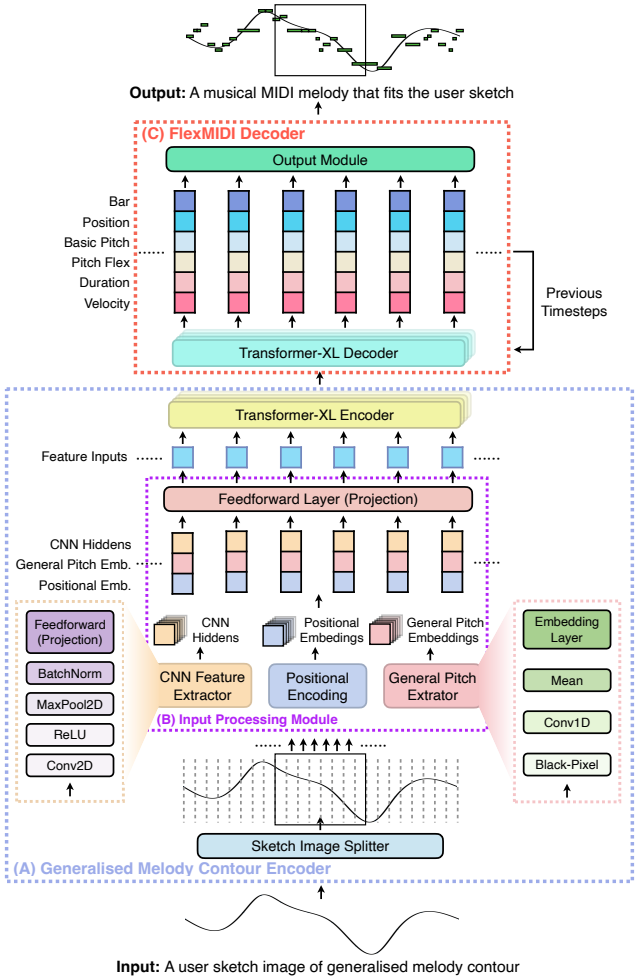

Fig. 2. The pipeline of Drawlody architecture. Firstly, the input image is split into sub-images, which are then processed by a CNN to extract relevant features. Next, a General Pitch Extractor and Positional Encoding are applied to further enhance the feature representation. The outputs of these three components are concatenated and linearly projected to create the feature inputs to the encoder. Conditioned on the input, the decoder then generates FlexMIDI symbols, which are converted into MIDI files by the output module.

B. Input Representation: Melody Contour and Generalised Melody Contour

A melody contour is the visual representation of pitch motions in its corresponding melody. It resembles a wavy curve that chronologically follows the pitch of each melodic note. Figure 3 displays three basic types of melody contour.

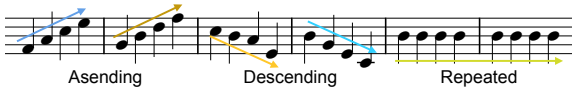

Fig. 3. Three basic types of melody contour (ascending, descending and repeated)

However, the raw form of a melody contour is complex: It forms a frequently undulating curve by strictly following each pitch in the melody (Figure 4A). Thus, using this complex curve as user input can be impractical: interaction-wise, drawing a detailed and wavy pitch curve is an overwhelming task for users. Algorithm-wise, raw melody contours can mislead generative models due to the presence of many intricate details, such as subtle changes and fluctuations in contours.

Taking the aforementioned problems into account, we introduce generalised melody contour (GMC), which omits the peripheral details of raw melody contours and describes the most fundamental pitch trend of a melody (Figure 4(C)). Compared to raw melody contours, GMC is far simpler in form and more concise to both machines and non-experts without a music background, greatly enhancing system usability.

C. Synthesising Generalised Melody Contour

The synthesis of a generalised melody contour includes two steps: (1) Basic melody extraction, which extracts the most fundamental pitch development in the melody; and (2) interpolation, where the identified basic melody notes are interpolated to form a smooth curve that reflects the general pitch motion of the melody. Figure 4 illustrates the stages in generalising a raw melody contour. It can be seen that the generalised melody contour aligns with the extracted basic melody (Figure 4(B)), and reflect the general trend of the original melody as well (Figure 4(C)).

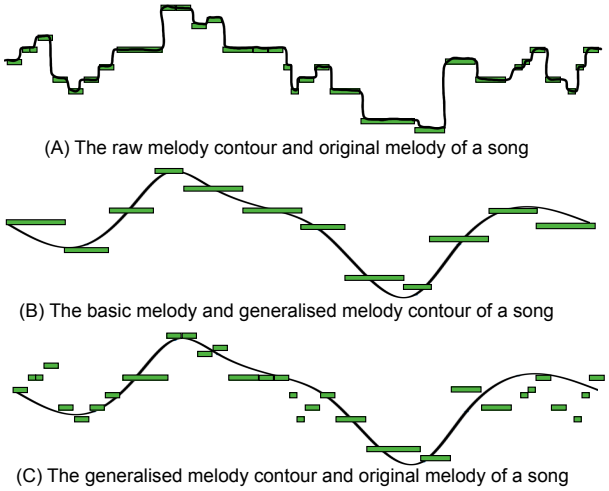

Fig. 4. Melody contour (A) and generalised melody contour (B, C). (A) follows each pitch in a melody and forms a wavy curve with many details, while (B) and (C) only follow the basic trend of a melody and remain simpler.

1) Basic Melody Extraction: Basic melody is an abstraction of the raw melody that retains only its most fundamental pitches (Figure 5B). This concept originates from Dai et al.'s research [14], where the authors select the most frequent pitch in each 2-beat segment of the original melody as its basic melody. However, this approach has certain limitations that result in failures during the extraction of basic melodies:

- **Outlier Failure**: It fails to process longer notes whose durations exceed 2 beats, and notes whose starts and ends fall in two different 2-beat segments. For example, the yellow frames in Figure 5 (A).
- **Same Frequency Failure**: It does not consider the case where all pitches in a 2-beat segment have the same frequency [1], for example, the blue frames in Figure 5 (A), where we need to establish priority rules to decide the most representative pitch.

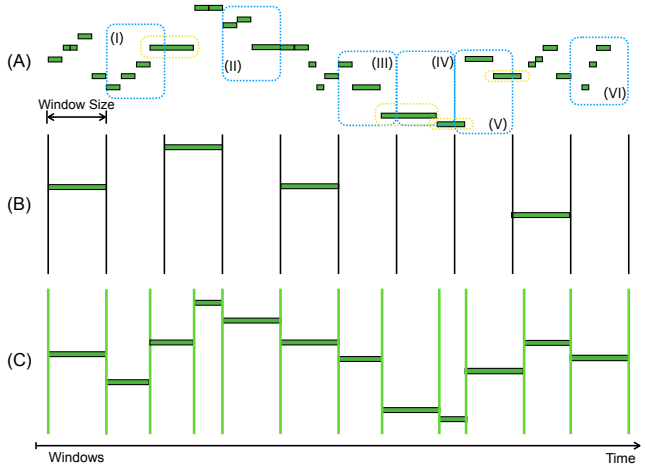

Fig. 5. Melody (A), original basic melody (B, [14]) and self-adaptively extracted basic melody (C, *ours*). The blue and yellow frames in (A) highlight the same frequency failures and outlier failures, respectively. The self-adaptive windows in (C) can handle longer notes and notes that fall out of the 2-beat size-invariable windows in (B), and also applies more priority rules to selecting the most representative pitches. For example, the outlier in (I) is a long note, so the algorithm splits another window to process this long note independently in (C); In (V), the outlier is a short note, so the algorithm resizes the window to incorporate this note.

Based on Schenker's theory [51] and some related work on the importance of melodic notes [61], [70], we propose a two-step self-adaptive basic melody extraction algorithm, leveraging music domain knowledge to process multiple note durations and their importance in deciding the melodic motion. The algorithm first employs a size-variable window to partition a melody into different segments. Then, within each window (segment), priority rules are applied to select the most representative pitch.

Self-Adaptive Melody Windowing: Technically, the initial size and stride of this window are two beats, following [14]. Then, the window begins to chronologically scan the melody notes from the left to the right. If the window encounters a long note (Equation 9) outlier, it automatically splits itself into two sub-windows to handle inliers and outliers separately. If a short note starts within the window but ends outside of it, the window resizes itself slightly to accommodate this short note. This self-adaptive windowing strategy addresses the outlier failure, enabling the algorithm to handle various melodies. More details of this algorithm are shown in a schematic graph in Section I of our supplemental materials.

Priority Rule-based Basic Pitch Detection: After defining how melody notes are grouped into segments by self-adaptive

[1]Note that frequency here does not refer to the frequency of a pitch. It means the number of times a pitch appears in a melody segment, i.e. the occurence frequency.

windows, we discuss the priority rules that the algorithm applies to decide the most significant pitch in each segment. Let $m_t^{(i)}.s$, $m_t^{(i)}.e$ and $m_t^{(i)}.p$ denote the start, end and pitch of a note $m_t^{(i)}$ in a melody $\mathbf{M}^{(i)}$, respectively, and $\mathbf{B}_s$, $\mathbf{B}_u$, $\mathbf{B}_w$ denote the strong, substrong and weak beats respectively:

- **In-tune Notes** refer to notes that fall in the scale of the melody's key.

$$N_\tau = \{m_t^{(i)} | m_t^{(i)}.p \% 12 \in \mathcal{S}\} \tag{4}$$

  where $\mathcal{S}$ represents the scale of the current key.
- **Off-tune Notes**, contrastly, refer to notes that fall out of the scale of the melody's key.

$$N_o = \{m_t^{(i)} | m_t^{(i)}.p \% 12 \in \mathcal{C} - \mathcal{S}\} \tag{5}$$

  where $\mathcal{C}$ is the chromatic scale. It is evident that both $N_\tau \cup N_o = \mathcal{C}$ and $N_\tau \cap N_o = \emptyset$ hold on these definitions.
- **Downbeat Notes** refer to notes starting from the strongest beats (namely, the first beat of a bar).

$$N_d = \{m_t^{(i)} | m_t^{(i)}.s \in \mathbf{B}_s; m_t^{(i)} \in \mathbf{M}^{(i)}\} \tag{6}$$

- **Substrong Beat Notes** refer to notes starting from the substrong beats.

$$N_u = \{m_t^{(i)} | m_t^{(i)}.s \in \mathbf{B}_u; m_t^{(i)} \in \mathbf{M}^{(i)}\} \tag{7}$$

- **Syncopations** refer to notes that are emphasised on a weak beat, but last till the next strong or substrong beat.

$$N_s = \{m_t^{(i)} | m_t^{(i)}.s \in \mathbf{B}_w \wedge m_t^{(i)}.e \in \mathbf{B}_s \cup \mathbf{B}_u; m_t^{(i)} \in \mathbf{M}^{(i)}\} \tag{8}$$

- **Long Notes** refer to the longest note within its temporal proximity ΔT (viz., a melody segment) [2].

$$N^+ = \left\{ m_k^{(i)} \,\middle|\, \begin{array}{l} k = \arg\max_t \left(\{(m_t^{(i)}.e - m_t^{(i)}.s)\} \right); \\ m_t^{(i)}.s \in \Delta T,\ m_t^{(i)} \in \mathbf{M}^{(i)} \end{array} \right\} \tag{9}$$

- **Short Notes** refer to the remaining non-long notes.

$$N^- = N^i - N^+ \tag{10}$$

where N^i denotes all note events in the melody $\mathbf{M}^{(i)}$.

Within each segment, the algorithm first filter out all off-tune notes. If there is only one note in the segment, or all notes are off-tune, the algorithm tweaks the pitch to be in tune along the pitch flow. Namely, if a note forms a decreasing (or increasing) pitch trend with its neighbouring notes, the algorithm shifts this pitch downward (upward). The algorithm finally picks one pitch following the priority chain shown in Equation 11, where notes with higher priority are considered more important in music theory [51].

$$N_\tau \cap N^+ \cap N_d \succ N_\tau \cap N_s \cap N^+ \succ N_\tau \cap N_d \succ N_\tau \cap N^+ \cap N_u \tag{11}$$

If there are multiple candidate pitches with the same priority, the algorithm favours the one with the highest frequency in the current window. If their frequencies are the same, the algorithm favours a higher pitch. Figure 5 compares a melody, its basic melody and its self-adaptively extracted basic melody.

[2]In this paper, we employed ΔT as two bars for 4/4 time music.

2) Interpolation: With a basic melody sequence of discrete basic notes, we first process these notes as paired data points $\{s_t, b_t\}_{t=0}^n$, where b_t and s_t denote the basic pitch value and its start. Then, a cubic spline interpolation algorithm is applied to these data points to obtain a smooth curve, resulting in the synthetic generalised melody contour. We chose cubic spline interpolation because it (1) is more suitable for mildly changing data points (like the basic melody) and the data magnitude of our model input; (2) can ensure that the output curve passes every data point smoothly, maximally sustaining faithfulness to the melody [23].

D. FlexMIDI Melody Representation

In the symbolic music domain, scholars have proposed many effective strategies to represent music in notation-based formats that can be parsed by computers, e.g. MIDI. Despite their impressive performance in many downstream tasks [44], [49], existing representation strategies use only MIDI note numbers to denote pitches in melodies, without considering the general melody trend information that is essential for our study. Therefore, we propose FlexMIDI representation, which integrates general melody trend information into symbolic music. Specifically, FlexMIDI uses two sequences to represent the pitch progression in a melody: (1) general trend information, namely the basic melody; and (2) pitch flex information. By introducing these two sequences, generative models can compose a basic melody trend while simultaneously elaborating its finer details with pitch flex. This facilitates the alignment between the shapes of GMCs and the generated melody trends, while still allowing for some level of aesthetic flexibility. The following parts will elaborate on the details of FlexMIDI.

1) Pitch Flex and Note Representation: Pitch flex is the signed interval between the actual pitches and the pitch of their basic melody note (basic pitch). The notion of pitch flex stems from the tonal analysis of music, where all melodic notes are perceived to be built **around a tonal centre** [58]. To simultaneously sustain the general trend and details of a melody, we treat notes in basic melody as local tonal centres in each segment (window), and use pitch flexes to describe the detailed melodic motion **around basic notes**, which forms a hierarchy shown in Figure 6. Technically, we use the same window as in self-adaptive basic melody extraction to scan a pair of melody and its basic melody from left to right. In each window, denote the pitches of all notes as $\{p_0^{(i)}, p_1^{(i)}, \ldots, p_k^{(i)}\}$ and the selected basic pitch in this window as b. The pitch flex sequence is defined as:

$$N_f = \{p_0^{(i)} - b, p_1^{(i)} - b, \ldots, p_k^{(i)} - b\} \tag{12}$$

To encode both general melody trend information and the details of melodies, FlexMIDI uses two symbols, Basic Pitch and Pitch Flex, to represent the pitch of a single melodic note. We also involve other two note-related attributes, Duration and Velocity, compressing these four symbols into a compound word that represents a note:

$$<\mathbf{BPitch}_b, \mathbf{Flex}_f, \mathbf{Dur}_d, \mathbf{Vel}_v> \tag{13}$$

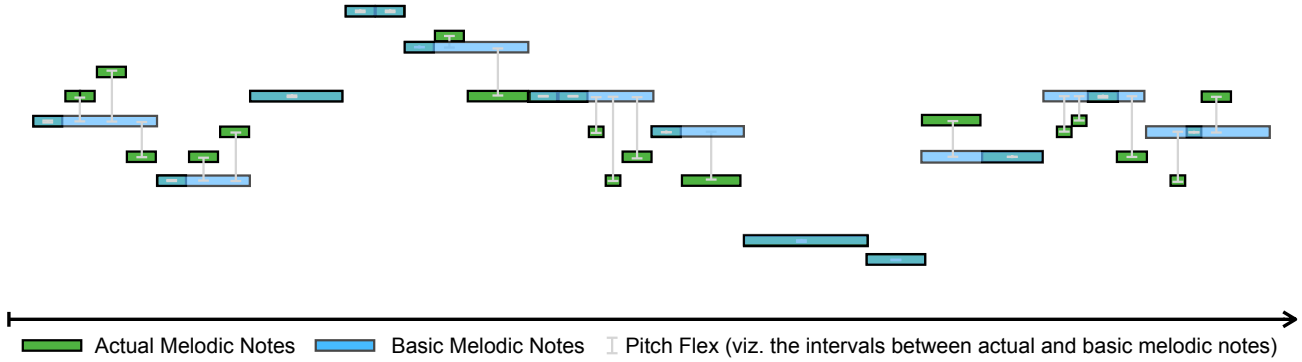

Fig. 6. An illustration of pitch flex, which is the interval between original (green) and basic (blue) melodic notes. In this figure, we overlap two different kinds of notes for an intuitive illustration of their relations.

where *b*, *f*, *d* and *v* indicate the actual value of basic pitch, pitch flex, duration and velocity, respectively. It is worth noting that the value of a duration symbol is determined by the type of its corresponding note. To exemplify, a $16th$ note is denoted by a $<\mathbf{Dur}_{16}>$ symbol, and a full note corresponds to a $<\mathbf{Dur}_{1}>$ symbol, by analogy.

2) Bar and Position Symbols: We use bar and position symbols [44] to notate the positional information of melody events. Specifically, a bar symbol represents the beginning of a new bar, followed by other symbols in that bar. A position symbol further elaborates a concrete position within that bar, followed by other symbols appearing at that position. Technically, we use $<\mathbf{Bar}_m>$ to represent the m-th bar, and $<\mathbf{Pos}_n>$ to denote the n-th location within the corresponding. In this paper, each bar is quantified as 32 equidistant positions, following [49]. The combination of bar and position symbols together marks the positions of their following symbols. For instance, the sequence $<\mathbf{Bar}_3><\mathbf{Pos}_{31}><\mathbf{BPitch}_{65}, \mathbf{Flex}_{-2}, \mathbf{Dur}_{16}, \mathbf{Vel}_{80}>$ notates a 16th note at the 31st position in the 3rd bar.

3) User-Specified Meta-Symbols: Users can customise the metadata of generated melodies with meta-symbols, which are special symbols that carry meta-information, such as the tempo (in beats per minute, BPM) and length (in bars) of the music. These meta-symbols are passed to the output module, which converts all symbols into MIDI files. The output MIDI files thus follow the meta-constraints given by users.

E. Model Architecture: Generalised Melody Contour Encoder

The overall architecture of Drawlody is displayed in Figure 2, where we employ an encoder-decoder model to guide sketch-based melody generation. The generalised melody contour (GMC) encoder takes a GMC sketch image and process it as high-dimensional conditional information. To handle GMC images of different sizes, the encoder first split a GMC image into a sequence of sub-images of the same size. After being processed by some feature extraction models, all sub-image features are serialised by a sequence model, to capture their dependencies. Specifically, the model utilises:

- a convolutional neural network (CNN), to extract features from GMC sub-images;
- an original general pitch extractor (GPE), to extract the general pitch information contained in GMC sub-images;
- a Transformer-XL (XFMR-XL), to capture the dependencies among different GMC sub-image.

These three parts reciprocally interact with each other. GPE refines a GMC image into a general pitch motion, while CNN extracts local details (e.g. rising trend or falling trend) from images and adds more flexibility to the pitch motion. The combination of GPE and CNN reduces the sequence length of input and simultaneously sustains multi-granularity features. Their hidden states at different timesteps are further serialised through a Transformer-XL with positional embeddings. Figure 7 displays the three key components of a GMC sub-image. The following subsections will describe the details of data preparation and model input.

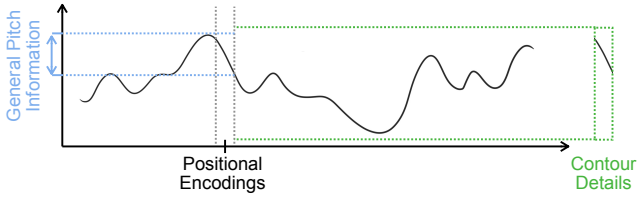

Fig. 7. Three key components of a GMC sub-image. The general pitch information reflects the overall pitch distribution of this sub-image; the contour details (e.g. falling trend) are indicated in the sub-image itself; the positional encodings inform the sequence model with the position of this sub-image.

1) GMC Image Split: Given a fixed-size sliding window $W \in \mathbb{R}^{s\times h}$, a GMC image $\mathbf{G}^{(i)} \in \mathbb{R}^{w\times h}$ is split into a chain of fixed-size sub-images $\mathcal{G} \in \mathbb{R}^{\lceil \frac{w}{s} \rceil \times s\times h}$, indexed by $\mathcal{T} = \{t | 0 \leq t \leq \lceil \frac{w}{s} \rceil - 1 \wedge t \in \mathbb{N}\}$, each corresponding to an input timestep. This can be formulated as a split function f_s:

$$\mathcal{G} = f_s(\mathbf{G}^{(i)}, W) = \{G_t^{(i)} \in \mathbb{R}^{s\times h} \ | t \in \mathcal{T}\} \tag{14}$$

where w is the width of $\mathbf{G}^{(i)}$; s denotes the width and stride of W; and h represents the height of both W and $\mathbf{G}^{(i)}$.

Then, we encode each sub-image $G_t^{(i)}$ into a vector $V_i \in \mathbb{R}^{d_{model}}$ with a CNN for feature extraction: $\mathbb{R}^{s\times h} \rightarrow \mathbb{R}^{d_{model}}$, where d_{model} denotes the dimension of Transformer-XL.

2) Extra Music-related Information: To allow for more controls from the user side, we also incorporate music-related information into inputs, such as general pitch information from GMC images, and user-specified metadata. Here, we mainly elaborate on the general pitch information extracted from GMC images.

In a Generalised Melody Contour (GMC) image, the contour is rendered black on a white canvas (e.g., see Figure 4B). The x-dimension (horizontal) of a GMC image represents time, while the y-dimension (vertical) represents pitch-related

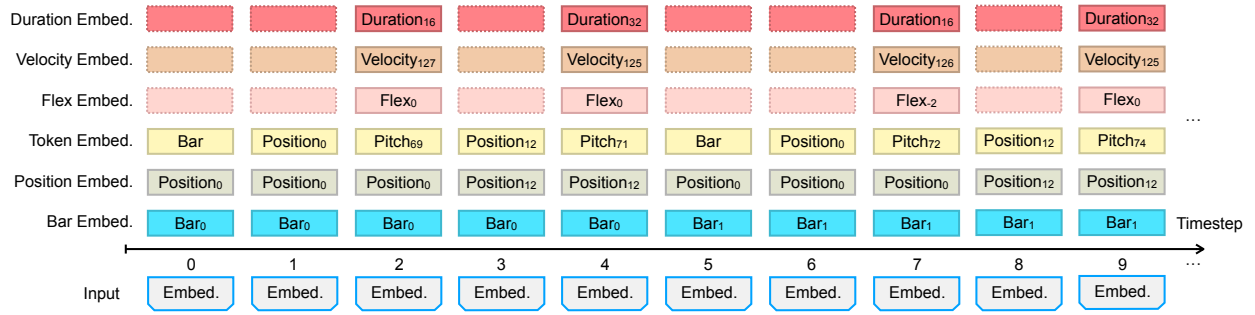

Fig. 8. The embedding overview of a FlexMIDI sequence $<\mathbf{Bar}_0><\mathbf{Pos}_0><\mathbf{BPitch}_{69},\mathbf{Flex}_0,\mathbf{Dur}_{16},\mathbf{Vel}_{127}><\mathbf{Pos}_{12}><\mathbf{BPitch}_{71},\mathbf{Flex}_0,\mathbf{Dur}_{32},\mathbf{Vel}_{125}><\mathbf{Bar}_1><\mathbf{Pos}_0><\mathbf{BPitch}_{72},\mathbf{Flex}_{-2},\mathbf{Dur}_{16},\mathbf{Vel}_{126}><\mathbf{Pos}_{12}><\mathbf{BPitch}_{74},\mathbf{Flex}_0,\mathbf{Dur}_{32},\mathbf{Vel}_{125}>$

information. However, capturing every black pixel in the image can result in redundant information, particularly when contours have higher contour thickness, leading to many adjacent pixel points. To address this issue, we propose a refinement technique for the y-coordinates of black pixels on the contour, aiming to capture only the most general pitch progression. This process reduces unnecessary details while preserving the essential melodic information. Specifically, we extract one general pitch for each GMC sub-image. General pitch refers to a representative pitch value (or y-coordinate in the image) that characterises the overall pitch distribution conveyed by a GMC sub-image. By obtaining a sequence of such general pitches, we effectively capture the pitch progression of the input contour.

Technically, we first scan the columns of a GMC sub-image $G_t^{(i)}$ and retrieve a position y_k for every single column c_k. y_k is determined by the mean of the y-coordinates of all black pixels in c_k:

$$y_k = \frac{\sum_{j=0}^{\nu} y_j}{\nu} \tag{15}$$

where ν is the number of black pixels within c_k. This step ensures that each column of the image has only one black pixel y-coordinate. We then collect the positions retrieved from all columns as $Y = \{(y_k)_{k=1}^{s}\}$, and the general pitch (p_g) information is the mean of 1D convolution of Y, which summarises the general pitch distribution within $G_t^{(i)}$. We build a dictionary that incorporates all possible values of p_g after rounding, which are mapped as embedding vectors.

3) GMC Feature Embedding: After the feature extraction from CNN and GPE, the final input to Transformer-XL forms a vector that incorporates (1) the melody contour information from CNN; (2) the pitch-related information, viz. the embeddings of tokenised events from the GPE; and (3) the relative positional embeddings [57]. We adopt a strategy similar to [49], concatenating three vectors and using a feedforward layer for dimensional reduction.

F. Model Architecture: Melody Decoder

The melody decoder takes the conditional information from GMC encoder, and decodes FlexMIDI symbols timestep-by-timestep. At the first timestep, a bar symbol $< \mathbf{Bar}_0 >$ is input as the proxy for the beginning-of-sequence symbol to initiate decoder generation. Then, the decoder outputs one FlexMIDI symbol at a timestep through the corresponding probability distribution over the dictionary.

1) FlexMIDI Sequence: Before being input to decoders, raw MIDI data are first quantised and processed as sequences of FlexMIDI symbols. Specifically, bar, position, and note symbols are chronologically ordered on the first hierarchy, while note symbols can be further expanded as basic pitch symbols, pitch flex symbols, velocity symbols, and duration symbols to form the second hierarchy. Figure 8 illustrates an excerpt of the FlexMIDI representation of an input MIDI item.

2) FlexMIDI Embedding: We use a dictionary to incorporate all FlexMIDI symbols. Each symbol is tokenised and embedded as a vector before being input to the decoder. At each timestep, if there is a basic pitch token, we first concatenate basic pitch and pitch flex embeddings as, which a linear layer further maps to a pitch embedding. Then, the embeddings of all symbols are concatenated as a longer vector, which is then mapped as a vector in the model dimension through another linear layer. Figure 9 shows the embedding strategy at a note symbol timestep.

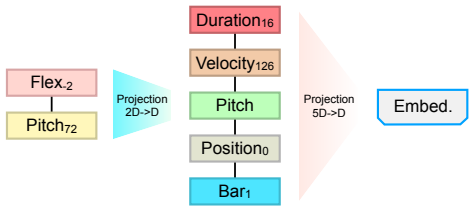

Fig. 9. The embedding strategy of a FlexMIDI note symbol. The pitch flex embedding (Flex_{-2}) and basic pitch embedding (Pitch_{72}) are first concatenated and linearly projected into a pitch embedding. This pitch embedding is then concatenated with other attribute embeddings, the output of which is projected as the final input embedding (Embed.).

3) Output Module: At each timestep, the hidden states of the decoders are processed through a softmax layer, which produces a probability distribution over the FlexMIDI symbol dictionary. The output module utilises beam search to sample FlexMIDI symbols from the dictionary and generate MIDI melodies based on the sampled Flex-MIDI symbols. It is important to note that when a note symbol is sampled, the module also samples all its attributes, namely basic pitch, pitch flex, velocity, and duration.

V. Experimental Setup

A. Dataset

1) Synthetic Dataset for Training and Validation: While the generalised melody contour (GMC) is a simple and understandable abstraction of melodic motions, manually collecting

GMC data can be a time- and energy-consuming task, especially on large datasets with long music samples. Inspired by the back-translation technique [43] from machine translation, we found that synthetic data is an effective strategy to alleviate the shortage problem of paired data. Therefore, we used the automatic pipeline described in subsection IV-C to instantly synthesise GMC representations from MIDI inputs.

In total, our dataset comprises 1,328,542 bars of MIDI melody data, equivalent to approximately 738 hours at a tempo of 120 bpm, all in 4/4 time and the C Major key, sourced from the Lakh LMD dataset [6]. This dataset includes 18,091 different songs in MIDI format, with the lengths of single MIDI files ranging from 15 to 499 bars, an average length of 48.58 bars, and a standard deviation of 30.11. We processed this dataset through the described pipeline (c.f. subsection IV-C), resulting in a collection of GMC-MIDI pairs used for both training and validation sets. More information on the dataset statistics is detailed in our supplemental materials.

2) Human Dataset for Testing: For testing, the model took human-sketched GMC data as input. If these sketches can still guide the model to generate coherent melodies, we deem the synthetic dataset effective.

B. Technical Implementation Details

The structure of Drawlody is shown in Figure 2. For the GMC encoder, we implemented a simple CNN architecture for image feature extraction and a General Pitch Extractor (GPE) for pitch-related feature retrieval. The extracted features are then input into a four-layer Transformer-XL [16] encoder after concatenation with relative positional encodings and a linear projection. The FlexMIDI Decoder consists of a four-layer Transformer-XL decoder and an output module for FlexMIDI decoding and off-tune note reduction. During decoding, we employed three different temperatures (0.8, 1.0, 1.2) to sample output tokens, and randomly selected melodies as the test data.

The Drawlody architecture was trained on three NVIDIA RTX A5000 GPUs, each with 20GB of memory. It took approximately 1.5 days for the model to converge on the loss function.

$$\begin{aligned}\mathcal{L}_{train} = \alpha\ell(\boldsymbol{z}_{Bar};\boldsymbol{\Theta}) + \beta\ell(\boldsymbol{z}_{Pos};\boldsymbol{\Theta}) + \gamma\ell(\boldsymbol{z}_{Token};\boldsymbol{\Theta}) \\ +\omega\ell(\boldsymbol{z}_{Flex};\boldsymbol{\Theta}) + \delta(\ell(\boldsymbol{z}_{Dur};\boldsymbol{\Theta}) + \ell(\boldsymbol{z}_{Vel};\boldsymbol{\Theta}))\end{aligned} \tag{16}$$

where α, β, γ, ω, δ are hyper-parameters, and ℓ is the cross-entropy defined as $\ell(\boldsymbol{z}_{\text{symbol}};\boldsymbol{\Theta}) = -\frac{1}{l}\sum_{t=0}^{l-1} z_{\text{symbol}}\log(p_t := P(m_t^{(i)}|m_{0:t-1}^{(i)};\ \mathbf{G}^{(i)};\ \mathbf{U}^{(i)};\Theta))$, for each input $\mathbf{G}^{(i)}$, $\mathbf{U}^{(i)}$, and the ground truth FlexMIDI symbol $\boldsymbol{z}_{\text{symbol}}$, where the symbol subscript indicates a specific type of symbol in FlexMIDI representation.

In the experiment, we adopted $\alpha = 0.1$, $\beta = 0.1$, $\gamma = 1.0$, $\omega = 1.0$, and $\delta = 0.85$. Here, γ, ω, and δ are parameters for token (FlexMIDI symbol) losses and token attribute (e.g. duration and velocity) losses, while α and β are losses for position information. We chose higher γ, ω, and δ than α and β, as the model was expected to optimise the token and token attribute losses more, which determine the shape of output melodies. We also found that slightly optimise $\ell(\boldsymbol{z}_{Bar};\Theta)$ and $\ell(\boldsymbol{z}_{Pos};\Theta)$ could already yield good results.

C. Music Quality Evaluation

1) Objective Music Quality Evaluation: For an objective analysis of the quality of machine-generated melodies, we used six computational metrics from [68], including four pitch-based features and two rhythm-based features. We calculated the overlapping area (OA) [68] between the feature values of the machine-generated samples and those of human-composed ones. **A larger OA** indicates a higher degree of similarity to human-composed melodies, which is deemed as being of **higher** quality by objective metrics. The pipeline of OA calculation is listed in Section VIII of our supplemental materials. For more details, readers may refer to the original paper [68].

Pitch-based Metrics

- **Pitch Count (PC)**: PC describes the number of different pitches within a melody sample;
- **Pitch Class Histogram (PCH)**: the octave-independent chromatic quantisation of the frequency continuum;
- **Pitch Class Transition Matrix (PCTM)**: a histogram-like matrix including pitch transitions for each ordered pair of notes. It implies useful information related to the melodic progression of music.
- **Average Pitch Interval (PI)**: the mean of intervals (in semi-tones) between two consecutive pitches in each melody sample.

Rhythm-based Metrics

- **Note Count (NC)**, which represents the number of different notes in a sample. Different from PC, NC excludes pitch information within notes and only focuses on rhythmic patterns.
- **Note Length Transition Matrix (NLTM)**, which, analogous to PCTM, implies useful information related to rhythmic patterns of melodies.

Faithfulness to the Sketch (F2S): We also proposed a metric named Faithfulness to the Sketch (F2S) score to assess the degree to which generated melodies align with their respective input sketches. This metric is based on the distance between the generated melody notes and the input sketch curve. Let all note events in a generated melody sample $\mathbf{M}^{(i)}$ be denoted as $N^{(i)} = \{m_0^{(i)}, m_1^{(i)}, \ldots, m_t^{(i)}\}$, and the input sketch as $\mathbf{G}^{(i)}$. The F2S score is determined by:

$$F2S = \lambda\frac{|N^{(i)}|}{\sum_{t=0}^{|N^{(i)}|-1}|m_t^{(i)}.pitch - \mathbf{G}^{(i)}(m_t^{(i)})|} \tag{17}$$

where $m_t^{(i)}.pitch$ denotes the pitch of the note $m_t^{(i)}$; $|N^{(i)}|$ is the length of $N^{(i)}$; $\mathbf{G}^{(i)}(m_t^{(i)})$ denotes the y-coordinate of the sketch curve at the start time of $m_t^{(i)}$; λ is a normalisation factor. In this paper, we selected $\lambda = 10^4$ considering the data magnitude. F2S measures the reciprocal of the average distance between the generated melody notes and the input sketch curve. A higher F2S score can indicate a closer sketch-melody alignment.

2) Subjective Music Quality Evaluation: We also conducted subjective tests to further assess the perceptual quality and the sketch-melody matching degree of generated samples. For each generated melody, we presented human participants with (1) the generated melody rendered in piano audio; (2) the

piano roll visualisation of this melody, overlapped with the input sketch. Participants were required to rate these shuffled samples on the following criteria:

- **Faithfulness** assesses the degree to which output melodies can match input sketches;
- **Stability** reflects the extent to which generated melodies can stably match user sketches over time;
- **Musicality** reflects the overall perceptual pleasantness of generated melodies;
- **Rhythmicity** measures how reasonable the rhythmic pattern of melodies sounds;
- **Richness** measures the overall aesthetic richness of generated melodies, instead of only rigidly fitting the input sketch or repeating some pitches, etc.

3) Sketch-based Baselines for Music Quality Evaluation: Previous studies on sketch-based melody composition [5], [37] only support fixed-length melody generation, or entail other musical conditions (e.g. chords, context melodies) as additional input. To make them directly comparable to Drawlody, we tweaked [36] and [37] as Rule-based and CNN-based baselines, respectively. We also propose a MIDI-based baseline as the contrast group to assess the effectiveness of FlexMIDI.

Rule-based (RB): This baseline is adapted from JamSketch [36], which uses a bi-gram-based genetic algorithm to generate music from human-sketched contours. Specifically, given a GMC image $G \in \mathbb{R}^{w \times h}$ and a pre-determined rhythm template $T_r = (s_i, e_i)$ where s_i and e_i mark the start and end of a note respectively, we split G into sub-images $\mathcal{S} = \{G_i \in \mathbb{R}^{w_i \times h} | w_i = e_i - s_i\}$ following T_r as boundaries. This rhythm template is pre-generated by another rhythm bi-gram trained on the same dataset. RB aims to generate one pitch for one sub-image using the fitness function Equation 18. We excluded the chord-dependent term from the original fitness function to make this baseline comparable to Drawlody:

$$F(N) = \omega_0 \mathbf{sim}(N) + \omega_1 \mathbf{seq}_1(N) + \omega_2 \mathbf{seq}_2(N) + \omega_3 \mathbf{ent}(N) \tag{18}$$

where $N = \{n_0, n_1, \ldots, n_{L-1}\}$ denotes a sequence of notes in the generated melody.

$\mathbf{sim}(N)$ measures the distance on y-coordinate (pitch) between notes and the input sketch curve:

$$\mathbf{sim}(N) = -\sum_{i=0}^{L-1} (n_i - g)^2 \tag{19}$$

where g is the y-coordinate of the pixel at the onset of n_i.

$\mathbf{seq}_1(N)$ represents the pitch bi-gram probability between two consecutive pitches:

$$\mathbf{seq}_1(N) = \sum_{i=i}^{L-1} logP(n_i | n_{i-1}) \tag{20}$$

$\mathbf{seq}_2(N)$ represents the interval bi-gram probability between two consecutive pitch intervals:

$$\mathbf{seq}_2(N) = \sum_{i=2}^{L-1} logP(n_i - n_{i-1} | n_{i-1} - n_{i-2}) \tag{21}$$

$\mathbf{ent}(N)$ measures the similarity between the entropy of all selected notes ($H(N)$) and that of a melody corpus (H_{mean}):

$$ent(N) = -(H(N) - H_{mean} - \epsilon)^2 \tag{22}$$

where ϵ is the controller of entropy, and lower ϵ tends to result in more complex melodies by influencing the entropy.

For more technicalities regarding these functions, readers may refer to the original paper [36]. In this paper, we had $\epsilon = 0$, $\omega_0 = 3$ and $\omega_1 = \omega_2 = \omega_3 = 1$, following [5].

CNN-based: This baseline tweaks the CNN-based JamSketch Deep-α [37] to fit our arbitrary-length and chord-free problem setting. We used an LSTM after CNN to handle inputs of different sizes and excluded all chord inputs.

D-MIDI and D-MuMIDI, which keep the same model structure as Drawlody, but use MIDI [3] and MuMIDI [49] during training and inference, respectively.

D-Raw, which uses raw (ungeneralised) melody contour data as input to train and validate the model.

D. Usability and Interpretability Evaluation

After the music quality evaluation, we also assessed the usability and interpretability of Drawlody against several existing interactive music creation systems. The usability evaluation is a human-grounded subjective evaluation [4], where participants needed to score the usability of each system on some metrics. For the interpretability evaluation, we employed both proxy-grounded evaluation and human-grounded evaluation from interpretable machine learning [20].

1) Proxy-Grounded Interpretability Evaluation: Proxy-grounded tasks interpret a black-box model against an inherently interpretable model, and study if there exist any similarities between them. We employed the rule-based baseline (RB) as the proxy model, which employs an interpretable genetic algorithm that step-by-step aligns the melody notes with the contour. For a similarity study, we compared objective feature scores and human music quality ratings to study similarities between their generated melodies.

2) Human-Grounded Evaluation of Usability and Interpretability: We also invited human participants to score the usability and interpretability based on some explanations of the generated music. For Drawlody, we presented participants with text- and visual-based explanations. Each visual-based explanation (e.g. Figure 10) is a music score consisting of (1) the generated melody; (2) the generated basic melody trend; and (3) the user-sketched contour. The text-based explanation further explains this visualisation, that "*Drawlody leverages the user-sketched contour to control the basic trend of the generated melody, meanwhile elaborating this trend into various pitches as the output melody to maintain coherence and flexibility*". For other baselines, we used texts to briefly describe their methodologies as explanations. For example, "*the chord-to-melody generation system learns the dependency between melodic notes and chords, such that the generated melodic notes are perceptually consonant with their accompanying*

[3] https://en.wikipedia.org/wiki/MIDI

[4] This study has been approved by the Department Ethics Review Committee (DERC) at the National University of Singapore under SOC-23-29.

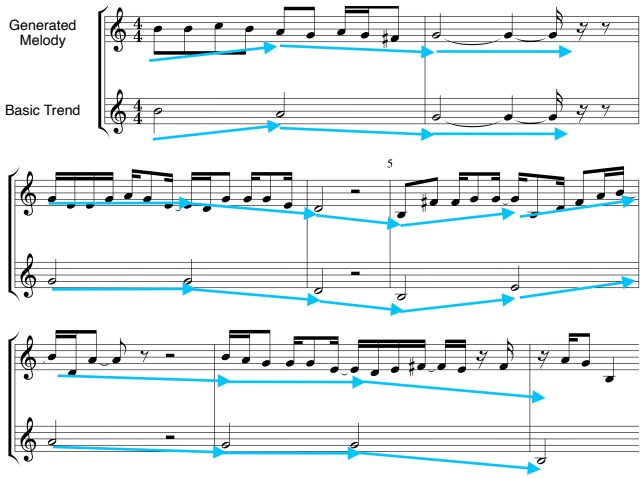

Fig. 10. Visual-based explanation of a melody generated by Drawlody. It can be seen that the basic trend (viz. the basic melody) aligns with the user-sketched contour (in blue), and the generated melody is an elaboration of the basic trend that also has a generally same direction as the contour.

chords". Based on these explanations [5], participants were asked to rate these systems on the following three metrics:

- **User Friendliness**: I feel the system is easy-to-use, without too much expert knowledge required.
- **User Engagement**: I feel I can be well engaged in the process of guiding generative models to generate music.
- **Interpretabillity**: From the given explanations, I can understand how my input guides melody generation.

3) Baselines for Usability and Interpretability Evaluation: We selected the following five interactive music creation systems as baselines for usability and interpretability evaluation.

Usability-Centred Baselines:

- **Emotion-based Music Generation (Emotion-based)** [72], which generates music from user-specified emotional styles.
- **Music Inpainting (Inpainting)** [5], which infills a short blank between two contextual measures with a pitch curve and a note density curve. We pre-set some default sets of contextual melodies to make this system not require any other music-related input, and classify this as a usability-centred baseline.

Interpretability-Centred Baselines:

- **Music Continuation (Continuation)**, an extension of [30] that continues writing user-given short melodies.
- **Chord-based Melody Generation (Chord-based)** [36], which generates melodies from user-given chord progressions.
- **Professional Digital Audio Workstation (DAW)**. We use Logic Pro 10.6.2 [6], a professional music production software widely employed in the industry.

E. Human Participants

We recruited 18 participants for all human-grounded studies, including 6 males and 12 females. 6 participants had some musical expertise with an average music training (or performing) experience of 8.5 years, while the remaining 12 are music lovers who listen to varied genres of music for more than 3 hours a day.

F. Ablation Study

To understand the contribution of the Convolutional Neural Network (CNN) and our General Pitch Extractor (GPE) to the entire Drawlody architecture, we propose three ablation variants of Drawlody, each corresponding to the removal of specific components:

w/o-CNN, where only the CNN architecture is removed;
w/o-GPE, where only the GPE is removed;
w/o-CG, where both CNN and GPE are removed.

G. Experimental Procedure

Participants were asked to complete the subjective music quality assessment first and then the usability and interpretability evaluation. They were allowed to take breaks during experiments to counterbalance possible influences of fatigues.

For the subjective music quality evaluation, each participant was asked to draw ten sketch curves following some instructive examples. Then, their sketches were pre-processed to smooth the stroke weight and keep their formats in accordance with our synthetic dataset. Melody samples were generated from these sketches through seven different models (including baselines and ablation variants) and shuffled with human-composed melodies (viz. ground truth) before being scored by the participants on a 5-point Likert scale.

VI. Results

A. Objective Music Quality Evaluation

Table I lists the objective scores of Drawlody and other baselines. The percentages represent the overlapping areas (OAs) between the objective score distributions of machine-generated melodies and those created by humans.

In terms of pitch-related metrics, while w/o-CNN narrowly outperforms Drawlody on PCTM, Drawlody exhibits the best overall performance with the highest OAs on PC, PCH, and PI compared to other groups. The advantage of w/o-CNN on PCTM (Pitch-Class Transition Matrix) may be attributed to its sole reliance on GPE for conditional input, which may tend to generate pitch transitions that are more similar to human-composed melodies after model optimisation.

Rhythm-wise, no significant differences in NC are observed among all FlexMIDI- and MuMIDI-based groups (viz., Drawlody, w/o-CNN, w/o-CPE, w/o-CG, CNN-based, and D-MuMIDI) due to their utilisation of bar-position rhythm representation. Despite this, FlexMIDI still surpasses MuMIDI on NLTM. This could be attributed to the involvement of basic pitch information, which considers the rhythmic importance of melody notes and can potentially guide the model to better capture the rhythm transitions. While the RB baseline achieves a similar OA on NC to FlexMIDI-based groups, its performance on NLTM remains the poorest. This is because RB uses fixed bi-grams learned directly from human-composed data, and is highly likely to exhibit high OA on NC, as NC only considers the number of different note lengths (e.g. 16th, 32nd notes, etc.). However, rhythmic bi-grams struggle

[5]More explanation examples are attached to our supplemental materials.
[6]https://www.apple.com/logic-pro/

TABLE I
THE OBJECTIVE ANALYSES OF DRAWLODY, BASELINES AND ITS THREE ABLATION VARIANTS. THE PERCENTAGES ARE THE OVERLAPPING AREAS (OA) BETWEEN THE SCORES OF MACHINE-GENERATED MELODIES AND HUMAN-COMPOSED GROUND TRUTH MELODIES. LARGER OAS MEAN HIGHER CLOSENESS TO HUMAN-COMPOSED MELODIES, WHICH IS DEFINED AS HIGHER QUALITY IN OBJECTIVE EVALUATIONS.

		Ablation Variants			Baselines				
	Drawlody	w/o-CNN	w/o-GPE	w/o-CG	Rule-based (JamSketch)	CNN-based (JamSketch Deep-α)	D-MIDI	D-Raw	D-MuMIDI
PC	**93.08%**	91.71%	87.07%	85.95%	79.60%	55.66%	63.12%	82.17%	61.65%
PCH	**96.30%**	95.55%	93.68%	91.79%	44.02%	90.43%	85.15%	89.38%	80.05%
PCTM	89.45%	**91.24%**	88.77%	86.55%	85.04%	89.33%	82.53%	60.52%	84.93%
PI	**83.18%**	76.13%	75.19%	60.06%	51.95%	76.73%	80.50%	60.72%	44.71%
NC	89.50%	88.99%	90.80%	89.88%	89.07%	88.86%	84.80%	43.45%	90.09%
NLTM	90.06%	89.29%	90.52%	89.49%	62.23%	92.06%	87.13%	60.28%	86.45%
F2S (w/ p-value)	**10.74**	9.22 (0.025)	9.91 (0.045)	6.65 ($1.02e^{-8}$)	30.67 ($2.10e^{-15}$)	5.95 ($1.51e^{-10}$)	3.63 ($9.53e^{-17}$)	5.56 ($9.55e^{-15}$)	8.91 (0.0011)

TABLE II
THE SUBJECTIVE ANALYSES OF DRAWLODY, BASELINES, AND ITS THREE ABLATION VARIANTS. MOSS REPRESENT THE MEAN OPINION SCORES ON EACH METRIC ON A 5-PT SCALE, AND THE P-VALUES ARE OBTAINED FROM MANN-WHITNEY U TESTS BETWEEN DRAWLODY AND OTHER MODELS.

	Drawlody	Ablation Variants						Baselines										Human (Ground Truth)	
		w/o-CNN		w/o-GPE		w/o-CG		Rule-based (JamSketch)		CNN-based (JamSketch Deep-α)		D-MIDI		D-Raw		D-MuMIDI			
	MOS	MOS	p-value	MOS	p-value	MOS	p-value	MOS	p-value	MOS	p-value	MOS	p-value	MOS	p-value	MOS	p-value	MOS	p-value
Faithfulness ↑	**4.21**	3.71	$2.73e^{-10}$	3.58	$3.94e^{-15}$	3.07	$1.07e^{-32}$	4.27	0.16	2.42	$1.54e^{-42}$	2.38	$1.64e^{-43}$	2.89	$3.68e^{-35}$	2.14	$1.02e^{-47}$	4.29	0.16
Stability ↑	**4.20**	3.70	$2.12e^{-10}$	3.59	$2.05e^{-13}$	2.88	$2.04e^{-35}$	4.12	0.30	2.64	$1.82e^{-37}$	2.59	$1.11e^{-34}$	2.72	$1.86e^{-37}$	2.13	$1.97e^{-46}$	4.36	0.025
Musicality ↑	**4.02**	3.31	$2.49e^{-16}$	3.40	$1.66e^{-12}$	3.16	$5.46e^{-22}$	2.33	$1.89e^{-41}$	3.36	$2.93e^{-14}$	3.40	$3.83e^{-12}$	2.35	$1.54e^{-41}$	2.28	$6.00e^{-41}$	4.22	0.0036
Rhythmicity ↑	**4.01**	3.26	$2.05e^{-15}$	3.39	$9.07e^{-13}$	3.22	$5.20e^{-19}$	3.01	$3.93e^{-25}$	3.23	$4.39e^{-17}$	3.01	$2.32e^{-23}$	2.51	$2.19e^{-38}$	2.76	$2.07e^{-32}$	3.88	0.10
Richness ↑	**3.71**	3.14	$1.16e^{-10}$	3.26	$3.04e^{-7}$	3.13	$4.56e^{-11}$	2.57	$4.23e^{-26}$	3.06	$4.87e^{-12}$	2.91	$3.17e^{-14}$	2.48	$1.47e^{-26}$	2.08	$9.59e^{-37}$	3.93	0.039

to handle long-term rhythm coherence over the entire song, thereby resulting in a notably low OA on NLTM.

Regarding faithfulness to the sketch, the RB baseline leverages a genetic algorithm to find the closest-to-curve pitch at each timestep, achieving the highest F2S score. However, this strict constraint often results in overfitting to the sketch and less pleasing melodies accordingly, as indicated by markedly lower scores on other metrics. In contrast, Drawlody relaxes the constraint with basic melody, allowing for more flexibility in generated melodies. This balanced approach ensures faithfulness to the sketch and musical pleasantness simultaneously.

B. Subjective Music Quality Evaluation

Table II presents analyses of subjective responses from both expert and amateur participants. For each metric, we calculated the mean opinion score (MOS) for each model, using Mann-Whitney U tests for statistical comparisons between models. From the overview, it is evident that Drawlody outperforms all its ablation variants and baselines significantly in terms of musicality, rhythmicity, and richness (with $p \leq 2.73e^{-10}$). This advantage becomes even more noticeable when comparing Drawlody to RB, D-Raw, and D-MuMIDI, all of which exhibit significantly lower MOSs on these three metrics.

For the RB baseline, its poorer performance can be attributed to two factors. First, RB uses two separate bi-grams to generate pitches and rhythms independently. This strategy (1) tends to lose the long-term conherence, as bi-grams only consider the coherence between two consecutive elements; (2) can result in dissonance between pitches and rhythms, as they are generated separately. Second, RB employs a genetic algorithm that selects pitches as close to the input curve as possible at each timestep (Equation 18). This may lead to overfitting to the input sketch, influencing the overall quality of the generated melodies.

Regarding D-Raw, the use of raw melody contour as input could introduce excessive pitch details that may mislead generative models (c.f. subsection IV-B). The model might struggle to capture the useful information (viz., the general trend), due to the noisy nature of raw melody contours. Similarly, D-MuMIDI does not consider the general trend information compared to Drawlody. This can make it challenging for the model to successfully capture the dependencies between input sketches and output melodies.

On rhythmicity, the advantage of Drawlody over other groups ($p \leq 3.04e^{-7}$) seems to somewhat belie the findings in objective evaluation, where there should be no significant disparities among FlexMIDI- and MuMIDI-based groups (namely, Drawlody, w/o-CNN, No-CPE, w/o-CG and CNN-based). This could be attributed to the influence of musicality on rhythmicity, where lower musicality might affect subjects' judgement of rhythmicity. This occurs because melodies and rhythms are integrated in human music listening, while objective metrics can only measure the rhythms of melodies by neglecting other pitch-related features. On faithfulness and stability, Drawlody has a remarkably close MOS to RB ($p \geq 0.16$). It also has similar faithfulness performance to human-composed melodies ($p = 0.16$). These advocate the ability of Drawlody to well fit the sketched curve while keeping the musicality.

Despite the generally positive performance of Drawlody, there are still gaps between Drawlody and human-composed melodies, particularly in terms of stability, musicality, and richness ($p \leq 0.039$). These differences indicate that there is still a considerable distance to cover before machine composers can truly compete with expert human musicians.

C. Usability and Interpretability Evaluation

1) Proxy-grounded Evaluation: For the proxy-grounded evaluation, we assessed Drawlody against the rule-based base-

TABLE III
THE USABILITY AND INTERPRETABILITY EVALUATION OF DRAWLODY. SINCE THE INPAINTING-BASED BASELINE HAS PROVIDED TWO CONTEXTUAL MELODIES AS THE EFAULT

		User Friendliness		User Engagement		Interpretability	
		MOS	p-value	MOS	p-value	MOS	p-value
	Drawlody	4.38	N.A.	4.63	N.A.	4.50	N.A.
Usability-centred	Emotion-based	4.13	0.5	3.88	0.03	2.00	0.04
	Inpainting	2.00	$2.0e^{-8}$	2.81	$4.1e^{-5}$	3.00	0.01
Interpretability-centred	Continuation	2.94	$3.0e^{-4}$	3.31	$3.1e^{-5}$	3.00	$8.0e^{-3}$
	Chord-based	2.94	$1.0e^{-4}$	3.31	$3.1e^{-5}$	3.00	0.1
	DAW	2.38	$2.8e^{-5}$	2.69	$4.1e^{-5}$	2.75	0.1

line (RB) as a proxy model. We investigated their similarity in both objective and subjective music quality scores. Regarding objective metrics, Table IV reveals a significant similarity in the pitch distributions generated by Drawlody and RB, with high Overall Agreement (OA) on Pitch Class (PC) at 90.79% and Note Content (NC) at 88.63%. Similarly, subjective metrics in Table II demonstrate similarities between Drawlody and RB in terms of faithfulness and stability ($p \geq 0.16$), two critical criteria associated with the sketch-melody alignment.

However, Drawlody does not exhibit substantial similarity with RB on other metrics. We attribute this difference to the discernible gaps in their music generation quality, as highlighted in the music quality evaluation mentioned earlier. These findings support the conclusion that while Drawlody can generally emulate the melodic motion depicted in sketches similar to RB, it introduces other adjustments to enhance the musicality of generated melodies, thereby avoiding overfitting to the input sketches.

TABLE IV
THE PROXY-GROUNDED EVALUATION OF DRAWLODY ON OBJECTIVE MUSIC QUALITY METRICS. THE PERCENTAGES ARE THE OVERLAPPING AREAS (OA) BETWEEN FEATURES SCORES OF DRAWLODY AND RB.

	PC	PCH	PCTM	PI	NC	NLTM
Drawlody v.s. RB	90.79%	56.90%	82.98%	71.33%	88.63%	85.44%

2) Human-grounded Evaluation for Usability and Interpretability: Table III lists human ratings on the usability and interpretability of some interactive music creation (IMC) systems. It is evident that Drawlody achieves similar MOSs to the usability-centred emotion-based system on User Friendliness and User Engagement, meanwhle significantly outperforming all other baselines that require musical conditions as input (viz. continuation, chord-based, inpainting, and DAW) by a large margin, with $p \leq 3.0 \times 10^{-4}$. This indicates that removing complex musical input can significantly enhance the usability of IMC systems.

Furthermore, Drawlody also achieves higher MOSs than both interpretability- and usability-centred baselines on Interpretability. This advantage is significant ($p \leq 0.04$) in contrast to all usability-based baselines and the continuation baseline. While the MOS of Chord-based and DAW is lower than that of Drawlody, the difference does not exhibit statistical significance ($p = 0.1$), as both of them emphasise interpretability and can give humans some hints from domain knowledge.

Generally, these results demonstrate our motivation that Drawlody can simultaneously retain the advantages of both usability-centred and interpretability-centred interactive music creation systems, and that the explanations given by Drawlody can help humans better understand how their input controls and relates to the generated melody. For more video demonstrations of generated melodies, the readers may visit this link.

VII. CONCLUSION

This paper presents Drawlody, an interactive music creation (IMC) framework with higher interpretability and usability. Based on a music-theoretical concept "*melody contour*", we create a simpler interaction medium generalised melody contour (GMC), balancing the trade-off between interpretability and usability in existing IMC frameworks. With Drawlody, users can compose arbitrary-length melodies by sketching simple GMC curves. The underlying architecture of Drawlody is based on CNN and Transformer, which generates melodies from GMC sketches following an end-to-end paradigm. We also propose new representations of melody contour images and symbolic music, corroborating their effectiveness through contrastive studies. The efficacy of Drawlody is assessed by both subjective and objective evaluations, which demonstrate its ability to produce high-quality musical output while also show a significant improvement in interpretability and usability. However, our work currently focuses on the generation of monophonic melodies. In the future, we aim to further extend the framework to encompass multi-track music scenarios, enhancing its capacity for diverse musical expressions and exploring exciting opportunities for commercial applications.

ACKNOWLEDGMENTS

This research is supported by Ministry of Education of Singapore (MOE-T2EP20120-0012).

REFERENCES

[1] Saleema Amershi, Dan Weld, Mihaela Vorvoreanu, Adam Fourney, Besmira Nushi, Penny Collisson, Jina Suh, Shamsi Iqbal, Paul N Bennett, Kori Inkpen, et al. Guidelines for human-ai interaction. In *Proceedings of the 2019 chi conference on human factors in computing systems*, pages 1–13, 2019.

[2] Chunhui Bao and Qianru Sun. Generating music with emotions. *IEEE Transactions on Multimedia*, pages 1–1, 2022.

[3] Théis Bazin and Gaëtan Hadjeres. NONOTO: A model-agnostic web interface for interactive music composition by inpainting. In *Proceedings of the Tenth International Conference on Computational Creativity, ICCC 2019, Charlotte, North Carolina, USA, June 17-21, 2019*, pages 89–91. Association for Computational Creativity (ACC), 2019.

[4] Alessandro Benedetto, Maria Concetta Morrone, and Alice Tomassini. The common rhythm of action and perception. *Journal of cognitive neuroscience*, 32(2):187–200, 2020.

[5] Christodoulos Benetatos and Zhiyao Duan. Draw and listen! a sketch-based system for music inpainting. *Transactions of the International Society for Music Information Retrieval*, 5:141–155, 11 2022.

[6] Thierry Bertin-Mahieux, Daniel P. W. Ellis, Brian Whitman, and Paul Lamere. The million song dataset. In *Proceedings of the 12th International Society for Music Information Retrieval Conference, ISMIR 2011, Miami, Florida, USA, October 24-28, 2011*, pages 591–596. University of Miami, 2011.

[7] Sebastian Böck, Florian Krebs, and Gerhard Widmer. Joint beat and downbeat tracking with recurrent neural networks. In *ISMIR*, pages 255–261. New York City, 2016.

[8] Ke Chen, Weilin Zhang, Shlomo Dubnov, Gus Xia, and Wei Li. The effect of explicit structure encoding of deep neural networks for symbolic music generation. In *2019 International Workshop on Multilayer Music Representation and Processing (MMRP)*, pages 77–84. IEEE, 2019.

[9] Keunwoo Choi, George Fazekas, and Mark Sandler. Text-based lstm networks for automatic music composition. *arXiv preprint arXiv:1604.05358*, 2016.

[10] Kyoyun Choi, Jonggwon Park, Wan Heo, Sungwook Jeon, and Jonghun Park. Chord conditioned melody generation with transformer based decoders. *IEEE Access*, 9:42071–42080, 2021.

[11] Matthew Clauhs. Songwriting with digital audio workstations in an online community. *Journal of Popular Music Education*, 4(2):237–252, 2020.

[12] Alexis Clay, Nadine Couture, Myriam Desainte-Catherine, Pierre-Henri Vulliard, Joseph Larralde, and Elodie Decarsin. Movement to emotions to music: using whole body emotional expression as an interaction for electronic music generation. In *12th International Conference on New Interfaces for Musical Expression, NIME 2012, Ann Arbor, Michigan, USA, May 21-23, 2012*. nime.org, 2012.

[13] Shayan Dadman, Bernt Arild Bremdal, Børre Bang, and Rune Dalmo. Toward interactive music generation: A position paper. *IEEE Access*, 10:125679–125695, 2022.

[14] Shuqi Dai, Zeyu Jin, Celso Gomes, and Roger B. Dannenberg. Controllable deep melody generation via hierarchical music structure representation. In *Proceedings of the 22nd International Society for Music Information Retrieval Conference, ISMIR 2021, Online, November 7-12, 2021*, pages 143–150, 2021.

[15] Shuqi Dai, Xichu Ma, Ye Wang, and Roger B Dannenberg. Personalised popular music generation using imitation and structure. *Journal of New Music Research*, 51(1):69–85, 2022.

[16] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime G. Carbonell, Quoc Viet Le, and Ruslan Salakhutdinov. Transformer-xl: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 2978–2988. Association for Computational Linguistics, 2019.

[17] Shangzhe Di, Zeren Jiang, Si Liu, Zhaokai Wang, Leyan Zhu, Zexin He, Hongming Liu, and Shuicheng Yan. Video background music generation with controllable music transformer. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 2037–2045, 2021.

[18] Sascha Disch and Bernd Edler. Frequency selective pitch transposition of audio signals. In *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 29–32. IEEE, 2011.

[19] Yizhuo Dong, Xinyu Yang, Xi Zhao, and Juan Li. Bidirectional convolutional recurrent sparse network (bcrsn): An efficient model for music emotion recognition. *IEEE Transactions on Multimedia*, 21(12):3150–3163, 2019.

[20] Finale Doshi-Velez and Been Kim. Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*, 2017.

[21] W Jay Dowling and Thomas J Tighe. *Psychology and music: The understanding of melody and rhythm*. Psychology Press, 2014.

[22] Mohit Dua, Rohit Yadav, Divya Mamgai, and Sonali Brodiya. An improved rnn-lstm based novel approach for sheet music generation. *Procedia Computer Science*, 171:465–474, 2020.

[23] S.A. Dyer and J.S. Dyer. Cubic-spline interpolation. 1. *IEEE Instrumentation & Measurement Magazine*, 4(1):44–46, 2001.

[24] Jeff Ens and Philippe Pasquier. Mmm: Exploring conditional multi-track music generation with the transformer. *arXiv preprint arXiv:2008.06048*, 2020.

[25] Gaëtan Hadjeres and Frank Nielsen. Anticipation-rnn: Enforcing unary constraints in sequence generation, with application to interactive music generation. *Neural Computing and Applications*, 32(4):995–1005, 2020.

[26] Donghong Han, Yanru Kong, Jiayi Han, and Guoren Wang. A survey of music emotion recognition. *Frontiers of Computer Science*, 16(6):166335, 2022.

[27] Don Hejna and Bruce R Musicus. The solafs time-scale modification algorithm. *Bolt, Beranek and Newman (BBN) Technical Report*, 1991.

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

[29] Zhejing Hu, Yan Liu, Gong Chen, and Yongxu Liu. Can machines generate personalized music? a hybrid favorite-aware method for user preference music transfer. *IEEE Transactions on Multimedia*, 25:2296–2308, 2023.

[30] Cheng-Zhi Anna Huang, Ashish Vaswani, Jakob Uszkoreit, Ian Simon, Curtis Hawthorne, Noam Shazeer, Andrew M. Dai, Matthew D. Hoffman, Monica Dinculescu, and Douglas Eck. Music transformer: Generating music with long-term structure. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

[31] Yu-Siang Huang and Yi-Hsuan Yang. Pop music transformer: Beat-based modeling and generation of expressive pop piano compositions. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 1180–1188, 2020.

[32] Hsiao-Tzu Hung, Joann Ching, Seungheon Doh, Nabin Kim, Juhan Nam, and Yi-Hsuan Yang. EMOPIA: A multi-modal pop piano dataset for emotion recognition and emotion-based music generation. In *Proc. Int. Society for Music Information Retrieval Conf.*, 2021.

[33] Shulei Ji and Xinyu Yang. Emomusictv: Emotion-conditioned symbolic music generation with hierarchical transformer vae. *IEEE Transactions on Multimedia*, 2023.

[34] Tianyu Jiang, Qinyin Xiao, and Xueyuan Yin. Music generation using bidirectional recurrent network. In *2019 IEEE 2nd International Conference on Electronics Technology (ICET)*, pages 564–569. IEEE, 2019.

[35] Youngmoo E Kim, Erik M Schmidt, Raymond Migneco, Brandon G Morton, Patrick Richardson, Jeffrey Scott, Jacquelin A Speck, and Douglas Turnbull. Music emotion recognition: A state of the art review. In *Proc. ismir*, volume 86, pages 937–952, 2010.

[36] Tetsuro Kitahara, Sergio Giraldo, and Rafael Ramírez. Jamsketch: Improvisation support system with ga-based melody creation from user's drawing. In *Music Technology with Swing: 13th International Symposium, CMMR 2017, Matosinhos, Portugal, September 25-28, 2017, Revised Selected Papers 13*, pages 509–521. Springer, 2018.

[37] Tetsuro Kitahara and Akio Yonamine. Jamsketch deep α: A cnn-based improvisation system in accordance with user's melodic outline drawing. In *Proceedings of the 4th ACM International Conference on Multimedia in Asia*, pages 1–3, 2022.

[38] Sandeep Kumar, Keerthi Gudiseva, Aalla Iswarya, Shilpa Rani, KMVV Prasad, and Yogesh Kumar Sharma. Automatic music generation system based on rnn architecture. In *2022 2nd International Conference on Technological Advancements in Computational Sciences (ICTACS)*, pages 294–300. IEEE, 2022.

[39] Edward W Large, Iran Roman, Ji Chul Kim, Jonathan Cannon, Jesse K Pazdera, Laurel J Trainor, John Rinzel, and Amitabha Bose. Dynamic models for musical rhythm perception and coordination. *Frontiers in Computational Neuroscience*, 17:1151895, 2023.

[40] Shuyu Li and Yunsick Sung. Transformer-based seq2seq model for chord progression generation. *Mathematics*, 11(5):1111, 2023.

[41] Jason Martineau. *Elements of Music*. eBook Partnership, 2021.

[42] Tashi Namgyal, Peter Flach, and Raul Santos-Rodriguez. Midi-draw: Sketching to control melody generation. *arXiv preprint arXiv:2305.11605*, 2023.

[43] Longshen Ou, Xichu Ma, Min-Yen Kan, and Ye Wang. Songs across borders: Singable and controllable neural lyric translation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 447–467, Toronto, Canada, July 2023. Association for Computational Linguistics.

[44] M Aqmal Pangestu and Suyanto Suyanto. Generating music with emotion using transformer. In *2021 International Conference on Computer Science and Engineering (IC2SE)*, volume 1, pages 1–6. IEEE, 2021.

[45] Ashis Pati, Alexander Lerch, and Gaëtan Hadjeres. Learning to traverse latent spaces for musical score inpainting. In *Proceedings of the 20th International Society for Music Information Retrieval Conference, ISMIR 2019, Delft, The Netherlands, November 4-8, 2019*, pages 343–351, 2019.

[46] Christine Payne. Musenet. *OpenAI Blog*, 3, 2019.

[47] Laure Prétet, Gaël Richard, Clément Souchier, and Geoffroy Peeters. Video-to-music recommendation using temporal alignment of segments. *IEEE Transactions on Multimedia*, 25:2898–2911, 2023.

[48] Rohit Rangarajan. Generating music from natural language text. In *2015 Tenth International Conference on Digital Information Management (ICDIM)*, pages 85–88. IEEE, 2015.

[49] Yi Ren, Jinzheng He, Xu Tan, Tao Qin, Zhou Zhao, and Tie-Yan Liu. Popmag: Pop music accompaniment generation. In *Proceedings of the*

28th ACM international conference on multimedia, pages 1198–1206, 2020.
[50] Sahreen Sajad, S Dharshika, and Merin Meleet. Music generation for novices using recurrent neural network (rnn). In *2021 International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICSES)*, pages 1–6. IEEE, 2021.
[51] Heinrich Schenker. *Free Composition: Volume III of new musical theories and fantasies*, volume 1. Pendragon Press, 2001.
[52] Gwenaelle C Sergio, Rammohan Mallipeddi, Jun-Su Kang, and Minho Lee. Generating music from an image. In *Proceedings of the 3rd International Conference on Human-Agent Interaction*, pages 213–216, 2015.
[53] Yi-Jen Shih, Shih-Lun Wu, Frank Zalkow, Meinard Muller, and Yi-Hsuan Yang. Theme transformer: Symbolic music generation with theme-conditioned transformer. *IEEE Transactions on Multimedia*, pages 1–1, 2022.
[54] Ian Simon, Dan Morris, and Sumit Basu. Mysong: automatic accompaniment generation for vocal melodies. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 725–734, 2008.
[55] Florian Thalmann, Geraint A. Wiggins, and Mark B. Sandler. Representing modifiable and reusable musical content on the web with constrained multi-hierarchical structures. *IEEE Transactions on Multimedia*, 22(10):2645–2658, 2020.
[56] T.J. Tsai, Daniel Yang, Mengyi Shan, Thitaree Tanprasert, and Teerapat Jenrungrot. Using cell phone pictures of sheet music to retrieve midi passages. *IEEE Transactions on Multimedia*, 22(12):3115–3127, 2020.
[57] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.
[58] G Vidyamurthy and Jaishankar Chakrapani. Cognition of tonal centers: A fuzzy approach. *Computer Music Journal*, 16(2):45–50, 1992.
[59] Josiah W. Smith, Orges Furxhi, and Murat Torlak. An fcnn-based super-resolution mmwave radar framework for contactless musical instrument interface. *IEEE Transactions on Multimedia*, 24:2315–2328, 2022.
[60] Dongjing Wang, Xin Zhang, Yao Wan, Dongjin Yu, Guandong Xu, and Shuiguang Deng. Modeling sequential listening behaviors with attentive temporal point process for next and next new music recommendation. *IEEE Transactions on Multimedia*, 24:4170–4182, 2022.
[61] Zihao Wang, Kejun Zhang, Yuxing Wang, Chen Zhang, Qihao Liang, Pengfei Yu, Yongsheng Feng, Wenbo Liu, Yikai Wang, Yuntao Bao, et al. Songdriver: Real-time music accompaniment generation without logical latency nor exposure bias. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 1057–1067, 2022.
[62] Shiqi Wei, Gus Xia, Yixiao Zhang, Liwei Lin, and Weiguo Gao. Music phrase inpainting using long-term representation and contrastive loss. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 186–190. IEEE, 2022.
[63] Pawan Whig, Arun Velu, and Rahul Ready. Demystifying federated learning in artificial intelligence with human-computer interaction. In *Demystifying Federated Learning for Blockchain and Industrial Internet of Things*, pages 94–122. IGI Global, 2022.
[64] Xixuan Wu, Yu Qiao, Xiaogang Wang, and Xiaoou Tang. Bridging music and image via cross-modal ranking analysis. *IEEE Transactions on Multimedia*, 18(7):1305–1318, 2016.
[65] Wei Xu. Toward human-centered ai: a perspective from human-computer interaction. *interactions*, 26(4):42–46, 2019.
[66] Wei Xu, Marvin J Dainoff, Liezhong Ge, and Zaifeng Gao. Transitioning to human interaction with ai systems: New challenges and opportunities for hci professionals to enable human-centered ai. *International Journal of Human–Computer Interaction*, 39(3):494–518, 2023.
[67] Zihan Yan, Chunxu Yang, Qihao Liang, and Xiang 'Anthony' Chen. Xcreation: A graph-based crossmodal generative creativity support tool. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, UIST '23, New York, NY, USA, 2023. Association for Computing Machinery.
[68] Li-Chia Yang and Alexander Lerch. On the evaluation of generative models in music. *Neural Computing and Applications*, 32(9):4773–4784, 2020.
[69] Yi-Hsuan Yang and Homer H Chen. Machine recognition of music emotion: A review. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 3(3):1–30, 2012.
[70] Kejun Zhang, Xinda Wu, Tieyao Zhang, Zhijie Huang, Xu Tan, Qihao Liang, Songruoyao Wu, and Lingyun Sun. Wuyun: Exploring hierarchical skeleton-guided melody generation using knowledge-enhanced deep learning. *arXiv preprint arXiv:2301.04488*, 2023.
[71] Jingwei Zhao, Gus Xia, and Ye Wang. Q&a: Query-based representation learning for multi-track symbolic music re-arrangement. *arXiv preprint arXiv:2306.01635*, 2023.
[72] Kun Zhao, Siqi Li, Juanjuan Cai, Hui Wang, and Jingling Wang. An emotional symbolic music generation system based on lstm networks. In *2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, pages 2039–2043. IEEE, 2019.
[73] Hua Zhu, Shangfei Wang, and Zhen Wang. Emotional music generation using interactive genetic algorithm. In *2008 International Conference on Computer Science and Software Engineering*, volume 1, pages 345–348. IEEE, 2008.

Qihao Liang received his B.Eng. Degree in Digital Media Technology from the College of Computer Science and Technology, Zhejiang University, P.R. China in 2022. Advised by Professor Ye Wang, he is currently pursuing a Ph.D. in Computer Science at the School of Computing, National University of Singapore. His research focuses on the intersection of sound and music computing, interpretable machine learning, and human-computer interaction.

Ye Wang (Member, IEEE) received the B.Sc. degree from the South China University of Technology, China, in 1983, the M.Sc. degree from the Braunschweig University of Technology, Germany, in 1993, and the Ph.D. degree from the Tampere University of Technology, Finland, in 2002. He is an Associate Professor with the Computer Science Department, National University of Singapore (NUS) and NUS Graduate School - Integrative Sciences and Engineering Programme (ISEP). He established and directed the sound and music computing (SMC) lab (smcnus.comp.nus.edu.sg). Before joining NUS, he was a member of the technical staff with the Nokia Research Center in Tampere, Finland, for nine years. His research philosophy is that technology should be developed for good - such as expanding access, increasing affordability, and improving quality of healthcare and education. Guided by this philosophy, he explored a new programmatic research agenda: cognitive neuroscience-inspired Sound and Music Computing for Human Health and Potential (SMC4HHP) in the past decade, and tried to address two big questions within his programmatic research agenda.1) How to enable users to discover their preferred music that satisfies clinical requirements for gait rehabilitation and exercise via music search, recommendation and generation? 2) How to leverage on the relationship between speech and singing to build applications for speech intervention, as well as for human health and potential (HHP) in general?