Journal of New Music Research

ISSN: (Print) (Online) Journal homepage: www.tandfonline.com/journals/nnmr20

Personalised popular music generation using imitation and structure

Shuqi Dai, Xichu Ma, Ye Wang & Roger B. Dannenberg

To cite this article: Shuqi Dai, Xichu Ma, Ye Wang & Roger B. Dannenberg (2022) Personalised popular music generation using imitation and structure, Journal of New Music Research, 51:1, 69-85, DOI: 10.1080/09298215.2023.2166848

To link to this article: https://doi.org/10.1080/09298215.2023.2166848

Published online: 24 Jan 2023.

JOURNAL OF NEW MUSIC RESEARCH
2022, VOL. 51, NO. 1, 69–85
https://doi.org/10.1080/09298215.2023.2166848

Routledge
Taylor & Francis Group

Check for updates

Personalised popular music generation using imitation and structure

Shuqi Dai[a], Xichu Ma[b], Ye Wang[b] and Roger B. Dannenberg[a]

[a]Carnegie Mellon University, Pittsburgh, PA, USA; [b]National University of Singapore, Singapore, Singapore

ABSTRACT

Many practices have been presented in music generation recently. While stylistic music generation using deep learning techniques has became the main stream, these models still struggle to generate music with high musicality, different levels of music structure, and controllability. In addition, more application scenarios such as music therapy require imitating more specific musical styles from a few given music examples, rather than capturing the overall genre style of a large data corpus. To address requirements that challenge current deep learning methods, we propose a statistical machine learning model that is able to capture and imitate the structure, melody, chord, and bass style from a given example seed song. An evaluation using 10 pop songs shows that our new representations and methods are able to create high-quality stylistic music that is similar to a given input song. We also discuss potential uses of our approach in music evaluation and music therapy.

ARTICLE HISTORY
Received 12 May 2021
Accepted 3 January 2023

KEYWORDS
Automatic music generation; algorithmic composition; stylistic music generation; music structure; music imitation; personalised music generation; music preference; music familiarity

1. Introduction

Music is a universal language that has many formalisations, but there is no one guiding theory for music composition. Moreover, each person has unique music preferences. Our goal is to create personalised music automatically by imitating one or more songs that are selected by a user. To truly create new works, it is important to go beyond simple variation. Instead, we must capture, model and simulate a musical style. Given a style model, we can generate new music with enough novelty to be enjoyable, yet still retain similarity to the original.

One motivation for our work toward personalised music is the creation of music for Music Therapy. In particular, Rhythmic Auditory Stimulation (RAS) (Thaut et al., 1996) is a useful neurological music therapy that helps people with Parkinson's disease to improve their gait performance and mobility. Millions suffer from this disease. However, music pieces that meet both the criteria for RAS treatment, such as beat strength and tempo, and a patient's music preferences are hard to find. Limited therapist time impedes the selection and use of RAS in spite of expected benefits. By creating customised music, we can produce any desired tempo and beat strength needed for therapy.

An intuitive approach to personalised music is to imitate the music styles of a person's favourite pieces. Functional Magnetic Resonance Imaging (fMRI) data has revealed that familiarity is a crucial factor for emotional music engagement (Pereira et al., 2011). A study of music choice also showed that familiarity with music positively predicts preference for songs, play lists, and radio stations (Ward et al., 2014). We hope to contribute to more enjoyable and effective RAS for Parkinson's disease patients by capturing and automatically imitating patients' favourite songs. In addition, automatic generation of music for music therapy can avoid the copyright issues and access to large corpora of popular music, especially in communities with limited resources and budgets.

While generating music for RAS therapy was the primary inspiration for this work, the goal of creating enjoyable music automatically, with the possibility of song imitation, tempo control, orchestration, and many other attributes, has many other applications. Examples include music in video games, where music is often responsive to the game state but rarely personalised, and background music for videos, where creators have constraints and preferences but lack the time or skill to compose and produce the music.

Recent practice in stylistic music generation mainly focuses on machine learning of general musical rules or style, which tends toward generic musical output with no support for personalisation. The number of favourite and familiar songs that a person can provide is insufficient for deep learning approaches, which need large amounts of training data. However, every piece of music has its own distinctive abstract qualities, for example, structure, melodic contour, rhythmic pattern, chord progression,

CONTACT Roger B. Dannenberg rbd@cs.cmu.edu

bass line pattern, etc. These abstract qualities might vary a lot from piece to piece, even within a music genre and in works by the same composer. By focusing on distinctive musical qualities within the constraints of general rules of music, we hope to create more enjoyable music.

To avoid the problem of 'smoothing over' distinctive attributes, and to further our goal of imitation, we need to 'learn' from a single song that we call the *seed song*. However, we cannot rely *completely* on the seed song because that would result in too much similarity. Often, an overly similar song sounds as if the original were being played with mistakes or else the original was simply plagiarised. This is perhaps related to the 'uncanny valley' phenomenon (Mori, 1970). In addition, the inverted-U model of preference for music (Berlyne, 1971; Chmiel & Schubert, 2017; Sluckin et al., 1983) suggests that the pleasantness (hedonic value) of a music piece increases as its novelty increases (familiarity diminishes), and will decrease once the novelty reaches a certain level (Figure 1). Therefore, it is important to control how similar the new song should be in order to achieve both stylistic similarity and creative differences. This is a new challenge for the field of automatic music generation.

Another significant issue in music generation is music structure, particularly at the level of phrases and sections. Longer-term structure was at the heart of many early works on music generation, but largely ignored in more recent generation systems based on neural networks and sequence learning. The resulting failure of many systems to exhibit interesting long-term behaviour has now been widely recognised and is receiving renewed attention. In this paper, we consider longer-term and higher-level music structure formed through section repetitions. For example a common pattern in popular music songs is ABABB, where letters stand for sections and repeated letters represent an approximate repetition of a section. In

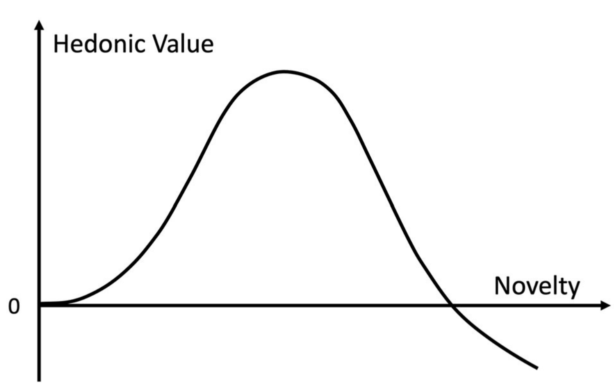

Figure 1. Inverted-U relationship: the Wundt Curve originally suggested by Wundt and later adapted by Berlyne (1971), and the linking of favourability to familiarity/time curve by Sluckin et al. (1983).

addition to repetition structure, there can be a common rhythm in A and B, a repetition in melody contour from one phrase to the next, etc. In this paper, we consider both higher-level repetition structure and lower-level repetition music pattern structure in our representation and generation process.

We introduce a stylistic music generation model that is able to capture melody, chord and bass style from a single pop song and imitate them with structure information in a new complete piece. Like 'music structure,' *music style* is a general term that can refer to almost any aspect of music. Definitions are further complicated by the multi-level, multi-modal character of music representation – music can be notated and read, performed, and listened to, and each of these modalities has aspects of style (Dai & Xia, 2018; Dannenberg, 1993). Style refers to general characteristics, but these can be seen at the level of music genre, sub-genre, composers, and individual compositions.

In this paper, *music style* refers to information at the symbolic music notation level, including rhythm, pitch and dynamics, and our work focusses on imitating single songs. Often, style at the level of a single song is *not* representative of more general styles, and some might not even use 'style' to refer to characteristics of a single song. Furthermore, there is no clear distinction between style at the levels of individual song and the more general genre level. For example, by imitating a Chinese popular song, we are at least likely to compose a mostly-pentatonic melody, and the result can be recognisable as Chinese popular music. Thus, song imitation is likely to discover and use *some* more general elements of musical style. We also note that imitation of performance, orchestration, and production (especially in pop music) are important aspects of style and perceived similarity, but we leave these to future work.

Our work focuses on popular music for multiple reasons. The original motivation for our work is to construct music for rhythmic auditory stimulation (RAS) for Parkinsons's disease patients, and therapists often use popular music. Secondly, most people have familiarity and knowledge of popular music styles. Therefore, it is easier to compare, discuss, and evaluate popular music than other music. It is also the case that popular music has many conventions that seem to simplify the music generation problem. This is not to say that truly great popular music is easy to make, but at least we can formalise approaches that generate *serviceable* popular music.

Our work offers three main contributions. First, we offer methods that generate likable music overall. If music is not likable, the fact that it is a successful imitation that listeners find *similar* to something likable is a small consolation. Second, we indeed produce music that listeners recognise as *similar* to our seed songs. Thus, we are able to learn enough from a single input song to form an imitation, even when seeds vary from Chinese Pop to Western Pop songs. Finally, one must ask if this whole enterprise of imitation to create likable songs is valid to begin with. By making imitatations, do we improve the preference for our machine-created songs? We will show through correlation that an increased preference for the seed song predicts an increased preference for our generated imitation. Thus, our original idea to create personalised music by imitation shows promise.

While our goal is *likable* music, evaluation necessarily relies on comparison and we often use the terms *preferred* and *preference*, making the assumption that *preferred* songs are more *likeable* and vice versa. While preference is good, note that we could create *preferred* songs simply by comparing to a very *unlikable* reference. To set a higher standard for success, our bases for comparison are human-composed hit songs, and we assume these are *likable* in an absolute sense. Our human raters are asked to indicate 'how much do you like' a given song, melody or other aspect of musical examples, at least indicating a rough sense of *likeability*.

Overall, we can say our methods generate music that listeners find likable, and we can use imitation without large training sets to enhance listener preferences for generated music. In addition, our methods can generate complete customised songs of any length and containing a logical, hierarchical music structure, as opposed to generating a few bars of music or long rambling sequences lacking in longer-term structure. We will also see how individual models for stylistic melody, chord and bass generation can be combined to create hybrid styles, e.g. creating a song with melody style from song A, chord style from song B, and bass style from song C. While we describe one particular system in detail, we believe there are many ideas that can inform future systems, serve as a baseline for comparison, and offer insights into music perception and cognition.

In the remainder of this paper, we first look at related or successful stylistic music generation models (Section 2), and then introduce the data preparation process (Section 3) and system design details (Section 4). In Section 5, we describe both objective and subjective experiments to evaluate the model output. We conclude in Section 6 with our findings and proposals for future research. More generated demonstrations can be found at https://www.shuqid.net/demo-smg.

2. Related work

Music theory, music grammars, automata and Markov models are widely used in automatic music generation (Conklin & Witten, 1995; Lo & Lucas, 2006; Thornton, 2009). Recently, deep generative models have become popular in this field (Briot et al., 2019). For stylistic music generation, current models mainly use the following approaches to model music style:

(1) Define style- and theory-related rules as constraints for music generation (Cope, 1991, 2000, 2005; Hiller et al., 1985). Rules make the generation fast and controllable; however, the results may lack creativity and therefore sound less interesting. Also, the quality of the model largely depends on the style/rule design, although quality can be quite high within a limited range of musical style. Since rules are often hand-coded, it is hard to extend or adapt rules to create new styles.
(2) Learn music style implicitly from a large amount of training data using deep learning models, especially sequence learning models (Hadjeres et al., 2017; Huang et al., 2017; Liang, 2016). While the results from this approach are impressive, they often lack common popular hierarchical music structure such as 4-or-8-bar phrases or repetition of melodic and rhythmic patterns. Recently, representation learning has motivated the practice of music style representation and disentanglement using Variational Auto-Encoders (VAEs) (Kawai et al., 2020; Roberts et al., 2018; Yang et al., 2019) and Generative Adversarial Networks (GANs) (Yu et al., 2017). These models are more controllable than previous deep learning approaches, e.g. VAEs can be used to interpolate between phrases or explore near neighbours along different learned dimensions. However, the latent space still remains unexplained, and these models can hardly generate complete music with hierarchical structure information. Often, the results are of fixed length.
(3) Encode long-term music structure information using the Transformer, a sequence model based on self-attention (Vaswani et al., 2017). This approach has been applied to stylistic music generation tasks (Hakimi et al., 2020; Huang et al., 2019; Payne, 2019). However, this approach requires a huge amount of training data and computation power, and so far has only been used for general music models because it would be difficult to find enough data to model a specific style. The Transformer architecture does not offer explicit representations of abstract music qualities such as chords, bass lines, or rhythms, so there is no straightforward way to control or bias the model to produce certain scales, hierarchical structure, repetition or even to limit repetition. This approach lacks explainability as well.

Elowsson and Friberg (2012) created an algorithmic composition system for popular music using probabilistic methods guided by music theory. Users can specify input settings like phrase length, chord transition matrix, and metrical salience histogram. The algorithm outputs a piece of music in MIDI format with melody, chord progression and simple phrases. They create a structure indicating phrase durations, accents within phrases, and phrase repetitions. Chords are then selected using a Markov chain. Melody generation is based on 10 different functions that estimate conditional probability of the next pitch and rhythm based on the previous note, current chord and accent locations, which are selected during the design of the overall structure. Humans rated the musical output slightly lower than human compositions, but not significantly. This system generates a generic popular music style, but cannot represent style variations, personalise music, or model bass style. We adapted some of the ideas in this work, for example, using theory and statistics to help capture and apply abstract qualities of style.

IDyOM (Pearce, 2005) is another probabilistic system based mainly on variable-order Markov models for adapting to and predicting musical sequences. IDyOM has been used to estimate information content of musical sequences, but it does not seem to contain sufficient constraints for its prediction to be used for music composition.

Evaluation of computer-composed music is often informal and ad-hoc. It is common to use human ratings based on short listening examples, and often comparisons are made to simple baseline models to show that improvement is achieved by new methods. Huang et al. (2019) use 180 comparisons to rate three techniques as well as excerpts from a human-composed dataset on the basis of 'which one is more musical' and using a Likert scale. Elowsson and Friberg (2012) compared machine- to human-composed songs along several dimensions using Likert scales. We believe ours is the first study to evaluate the degree to which computers can successfully compose music by imitation.

3. Data representation and pre-processing

The input/output data is represented in quantised MIDI format. We ignore everything but notes and meta-data such as tracks and instruments. To simplify our system, we follow the approach of Elowsson and Friberg (2012), which transposes everything to the key of C major or A minor, eliminating all accidentals (sharps or flats). For this study, we consider only songs in major mode. This is too limited to represent many pop songs accurately, but the reduced vocabulary improves the quality of learning and statistical summaries. In compositions, accidentals can truly sound like accidents if not used carefully, so ignoring them simplifies the task of composition.

Some critical music information such as structure and chord progression cannot be explicitly expressed in MIDI. Therefore, we convert MIDI to and from a data representation shown schematically at the left side of Figure 2. We restrict music to a time signature of 4/4, and the sixteenth note is the smallest unit of time, thus each bar contains an array of 16 sixteenth notes. For computations, we represent pitch as an integer in $[1 \ldots 15]$ representing the scale degree over 2 octaves, starting on C. Therefore, differences in pitch, e.g. $p_1 - p_2$ are diatonic intervals; e.g. we do not distinguish major from minor intervals.

We represent one chord for each bar, using a limited vocabulary of seven triads: Cmaj, Dmin, Emin, Fmaj, Gmaj, Amin, Bdim (I, ii, iii, IV, V, vi, vii°). This is certainly not a complete representation of harmony for pop music, but Elowsson and Friberg (2012) used this representation (minus vii chords) with good results. Chord

Figure 2. Data representation and its corresponding music notation. Here, integer arrays (of length 32) representing melody, chords, bass and drums as described in the text have been translated directly to ASCII strings (at left, also of length 32 plus added vertical bars to group measures of 16th notes) to illustrate our music encoding.

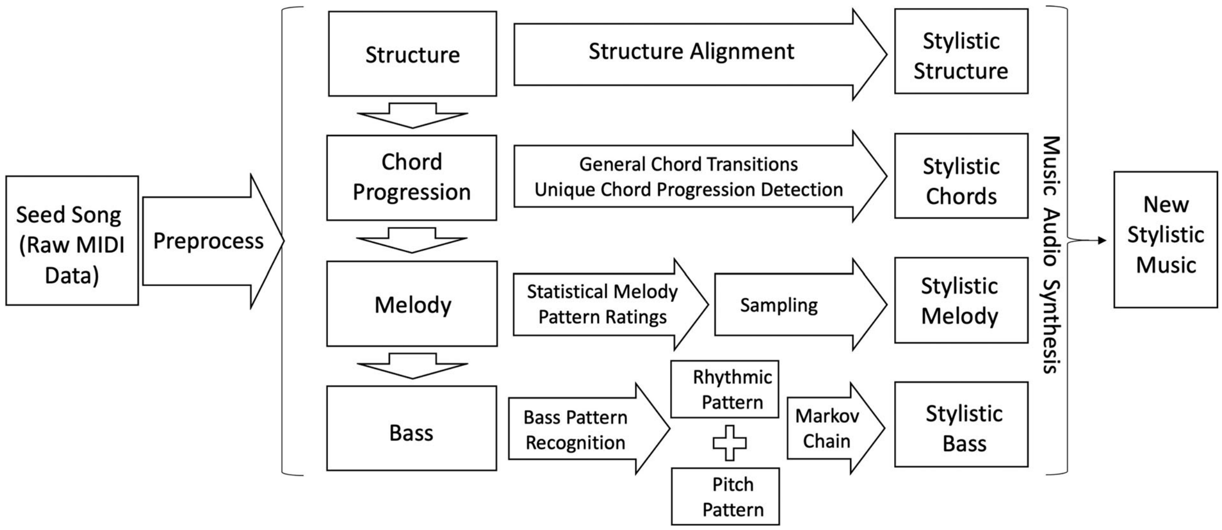

Figure 3. Framework of stylistic music generation system.

inversions are not distinguished here, but are implied by the bass. If the melody plays a 7th or other non-chord tone, it may imply a seventh or other chord type, but again, this is not directly represented in our chord vocabulary.

Each duration is represented by an integer in the range [1…16] denoting the number of 16th notes. Thus 'notes' are pairs of pitch and duration. In our representation, there are only 240 (15 pitches×16 durations) different notes, which makes it practical in music generation to estimate probabilities for every possible note.

We have developed systems to analyse music MIDI files automatically by extracting melody, chords, bass and structure (Dai et al., 2020; Jiang & Dannenberg, 2019), but for work reported here, we performed analyses by hand. We collected 13 MIDI songs and labelled them with an accurate analysis. While there is certainly some degree of subjectivity in labelling (Müllensiefen et al., 2007), our representation is more that of a simplified lead sheet than a detailed analysis, so label ambiguity is unlikely to have significant impact.

To deal with the limitations of our representation, we quantise all durations to 16^{ths}, remove accidentals (e.g. F♯ becomes F, B♭ becomes B, etc.) and remove chord extensions and inversions (Amin7/C becomes Amin, etc.). In rare cases, we must change minor chords to major or vice versa, e.g. a Dmaj becomes Dmin. When there is more than one chord per measure, we take the chord with the longest duration, with a preference for the first. While these substitutions are at best less interesting than the original, they rarely sound like outright mistakes because the result is simple and diatonic, and they occur rarely enough that they have little effect on statistical analyses.

4. Method

In this section we describe how we model and generate music. We begin with the overall model framework and then illustrate each module in the system individually.

4.1. Framework

We use statistical machine learning methods to imitate the styles of melody, chord progression, bass and structure from an input seed song. The system framework is shown in Figure 3. The system takes one seed song as input. After some data pre-processing steps (described in Section 3), we feed each part from the seed intoa corresponding generation module.[1] We also feed the results from structure and chord progression modules into the other modules as inputs. The tempo is set to the same as the tempo of the seed song input. Finally, we combine the newly generated stylistic melody, chord accompaniment and bass MIDI tracks and synthesise them to obtain audio output. The details for each module are described below.

4.2. Structure alignment and generation

We now discuss the higher-level structure of song sections. We describe structure using a list of tuples, where each tuple consists of a name (same sections or similar sections with slightly different variations share the same name), length (number of bars), and whether the section is a variation of the first section with the same name or not ('yes' means it is a variation, 'no' means it is exactly

[1] Seed songs for different modules can be different, e.g. taking the melody from seed A and bass from seed B.

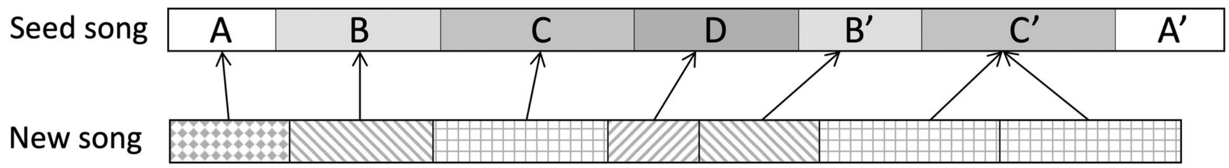

Figure 4. An example of ideal structure alignment.

the same section). For example, structure [[A, 8, no], [B, 10, no], [A, 8, no], [B, 10, no], [B, 10, yes]] means the song is divided into five sections 'ABABB.' The first section is named 'A' and has 8 bars; the last section is a variation of second section (named 'B').

We developed three ways to generate new structures: (1) copy the seed song structure; (2) generate according to a specification string such as 'AABABC' from the user and treat each section as 8 bars; (3) generate random structures, which includes selecting from a collection of typical structures. This gives flexibility in generation. Since the new structure can be different from the structure of the seed song, we need to align each new section to an appropriate seed song section. For example, in Figure 4, we want to imitate the seed song shown at the top using the new structure shown below it. A good alignment will imitate the seed's intro style in the new intro, imitate seed's first chorus in the new chorus, etc.

Since different songs have various structures and there is no standard solution for alignment, we deal with the problem in an unsupervised way using a 'greedy' heuristic approach. We extract features (L, P, V, F) from each section, representing length in bars, number of sections before this one, whether it is a variation section or not, and the number of sections with the same section letter. We define the distance between two sections as follows, where (L_1, P_1, V_1, F_1) describe a section from the seed song, (L_2, P_2, V_2, F_2) describes a section from the new song, L_s and L_n are the total bar lengths of seed song and new song, and P_s and P_n are the number of sections in the seed song and new song. We compute distance in terms of the four features L, P, V and F, weighting the feature differences by w_x, where $w_l = 0.7 > w_p = 0.15 = w_v = 0.15 > w_f = 0.1$, and $|x|$ is the absolute value of x. (Later, we also use $|x|$ to denote set cardinality).

$$distance = w_l\left|\frac{L_1}{L_s} - \frac{L_2}{L_n}\right| + w_p\left|\frac{P_1+1}{P_s} - \frac{P_2+1}{P_n}\right| + w_v\left|V_1 - V_2\right| + w_f\left|\frac{F_1}{P_s} - \frac{F_2}{P_n}\right| \tag{1}$$

We use the nearest section in the seed (according to *distance*) for each new song section. However, if a new song section is within an arbitrary threshold for *distance* to any of the seed song sections, we first try to align it with previous sections in the new song, choosing the one with the least *distance*. If no previous section is similar (below threshold), then we consider this section as a completely new section with no seed reference.

4.3. Stylistic chord generation

To generate convincing chord progressions while imitating the harmonic style of the seed song, we combine statistical features from a general popular music data set[2], seed song statistics, and distinctive sequences from the seed song. Within each section, we generate chord progressions in time order, selecting one or more successive chords at each step. Section endings are treated specially so that we can impose a stylistically consistent harmonic resolution to the section.

Blending transition matrix. Ignoring distinctive sequences for the moment, we create first-order Markov chain chord transition matrices from both general statistics and the seed song, then blend them together using $P_{trans} = (1-\alpha)P_{general} + \alpha P_{seed}$, where α is the tuning parameter. The combined transition probabilities are used to select the next chord up to the section ending sequence.

Distinctive chord sequences in the seed song. Distinctive chord sequences are an important aspect of harmonic style and give songs a recognisable sound. We identify distinctive chord sequences in the seed song by comparing statistical features between the seed song and a general dataset consisting of 300 annotated pop songs from github.com/tmc323/Chord-Annotations. As shown in Figure 5(a), we find all of the 2,3,4-gram chord sequences in the seed song that occur less than a threshold (5%) in the general statistics. We then extend our Markov chain approach (Figure 5(b)) by adding these chord sequences as possible new states (Figure 5(c)) resulting in an irregular-shaped transition matrix where some chords can transition to a short sequence. When such a transition occurs, the entire short sequence is output and generation continues from the last chord of that sequence.

[2] https://github.com/tmc323/Chord-Annotations.

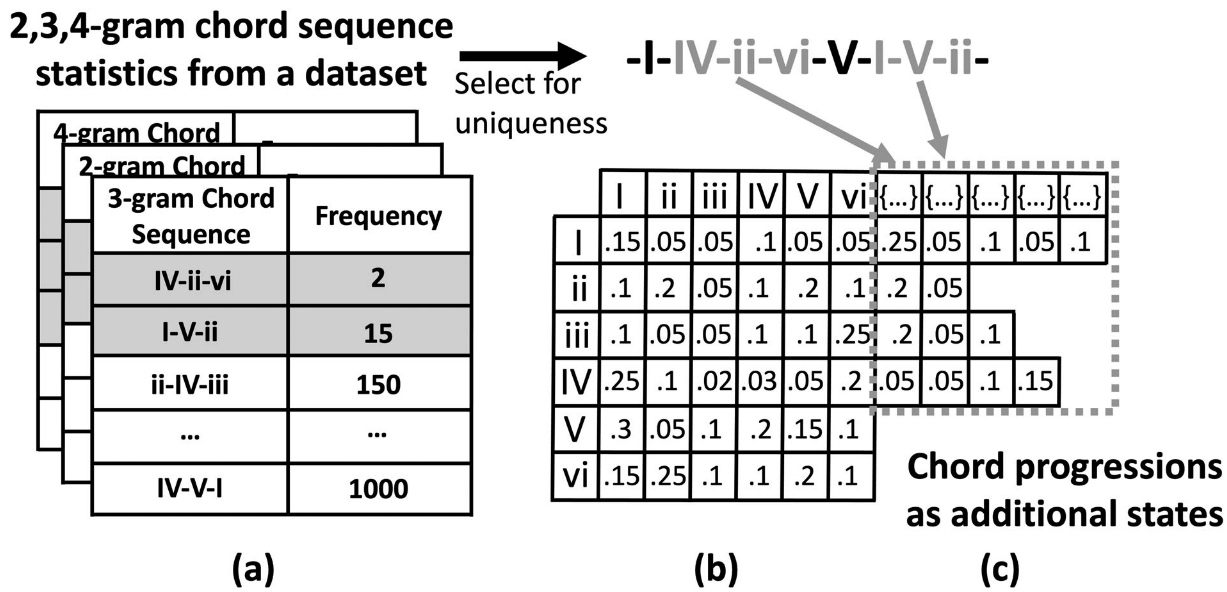

Figure 5. Distinctive chord sequence detection and generation, using Roman Numeral chords in major mode as example.

Stylistic cadence. Cadences are commonly used in popular music to end a phrase or section. Much like the handling of distinctive chord sequences, we calculate a cadence transition matrix that transitions from a chord to all cadence candidates (varying from a length of 2 chords to 6 chords). Beginning 6 chords from the end of a phrase, cadence sequences are chosen with a probability corresponding to their frequency in our dataset. If a cadence sequence is not chosen, a transition to a single chord is selected using normal transition probabilities.

4.4. Stylistic melody generation

From music theory, we know that rhythm and pitch interact to form melody. Thus, we designed a number of melody style rating functions to evaluate the suitability of the next note given previous notes in the context of a chord progression. We generate melody in time order, note-by-note, using rating functions to estimate the probability of each possible next note and making a weighted random choice. Thus, we use a statistical sampling method. To avoid outliers, we generate 30 candidate melody sequences and pick the one with best rating as the stylistic melody output.[3]

More formally, we have $n = 15$ rating functions (described below in this subsection) $P_i(x, y)$ where x is the next note (pitch and duration pair), and y is the context including the note position in the current bar and phrase, the previous notes, the chord progression, and the seed song. We treat these functions as independent probabilities, multiplying them together to form a weight for each note:

$$w_x = \prod_{i=1}^{n} P_i(x, y) \tag{2}$$

Finally we select the next note at random based on the weights w_x. Note that it is not necessary for the weights produced by the P_i to sum to one, and they are not all true probability functions. In the definitions that follow, we omit the usual normalisation to produce proper probability functions since it is unnecessary. Note that if a rule does not apply, P_i can just return the same value for all x so that P_i exerts no influence. On the other hand, a rule can completely disallow x by giving it a weight of 0.

The stylistic rating functions used to extract and represent melody style feature are inspired by Elowsson and Friberg (2012), who use similar methods to generate popular music. We extended these functions and use seed song statistics to encode specific melody style qualities from the input seed songs, and we developed additional rating functions that emphasise long-term structure. The rating functions consist of four categories: (1) rhythm-only ratings; (2) pitch-only ratings; (3) pitch and rhythm ratings; (4) long-term ratings.

Similar to chord transition matrix blending in the previous discussion, almost all of the rating functions blend general song statistics with seed song statistics, and we use α to represent the blending tuning parameter. Large α accentuates imitation at the risk of relying on very sparse (single seed song) data, while small α emphasises more reliable statistics, but reduces the degree of imitation.

[3] In our experience, statistical sampling often leads to very unlikely selections that seem to be musical mistakes, while making the most likely choice at every step can become highly predictable and repetitive. Generating many pieces and picking the most likely one rejects most songs with bad choices while preserving variety and an interesting degree of surprise.

Figure 6. Example of Original seed song melody (tune from Yankee Doodle), imitated melody generated using $\alpha = 0.5$ and $similarity_{rhythm} >= 0.8$, and non-imitated melody (using background general statistics) generated with same chord progression but $\alpha = 0$.

Before presenting the rating functions in detail, Figure 6 shows the effect of α in imitations of a familiar tune using the original chords. The first imitation, using $\alpha = 0.5$, is quite similar in pitch distribution and rhythm, while the second imitation, using $\alpha = 0$ is very different from the seed song. Both of the generated melodies are musical.

4.4.1. Pitch style rating functions

In the following, the notation |*set description*| means the cardinality of the described set. In general, we give details for probabilities based on the seed song, and we leave it to the reader to infer the formulas for *general* probabilities, which are computed in the same way, but use data from our entire collection of songs.

Pitch frequency. Frequency of pitch p occurring in both the seed song statistics and general popular music statistics:

$$\begin{aligned} P_{freq}(p) &= \alpha P_{freq,seed}(p) \\ &\quad + (1-\alpha) P_{freq,general}(p), \end{aligned}$$

$$\text{where} \quad P_{freq,seed}(p) = \frac{|\text{notes with pitch } p \text{ in seed song}|}{|\text{notes in seed song}|} \tag{3}$$

Pitch harmony with chord progression. Given chord c, probability of pitch p occurring in both the seed song statistics and general popular music statistics:

$$\begin{aligned} P_{har}(p|c) &= \alpha P_{har,seed}(p|c) \\ &\quad + (1-\alpha) P_{har,general}(p|c), \end{aligned}$$

where

$$P_{har,seed}(p|c) = \frac{|\text{notes with pitch } p \text{ and chord } c \text{ in seed song}|}{|\text{notes with chord } c \text{ in seed song}|} \tag{4}$$

Pitch interval frequency. Frequency of pitch interval $\Delta p = p - p_{prev}$ occurring in both the seed song statistics and general popular music statistics.

$$\begin{aligned} P_{interval}(\Delta p) &= \alpha P_{interval,seed}(\Delta_p) \\ &\quad + (1-\alpha) P_{interval,general}(\Delta_p), \end{aligned}$$

where

$$P_{interval,seed}(\Delta_p) = \frac{|\text{interval } \Delta_p \text{ in seed song}|}{|\text{intervals in seed song}|} \tag{5}$$

Pitch interval with harmony. Given chords c_1 and c_2, probability of the pitch interval $p - p_{prev}$ occurring in both the seed song statistics and general popular music statistics. This function is hand-crafted to capture multiple heuristics: (1) larger intervals are more likely to move to pitches that are more consistent with the chord, (2) larger intervals are also more likely to begin and end on chord tones, (3) larger intervals tend to go up, while smaller intervals tend to go down, (4) smaller intervals are more likely overall. We use $P_{har}(p|c)$ (Equation (4)), derived from song statistics, to determine pitches that are consistent with chords, but we use rules to express interaction with intervals.

Downbeat pitch harmonic. If a note starts on beat 1 or 3, then we add more dependency on the harmony. Given chord c for note with pitch p and duration d starting on beat 1 or 3, the harmonic rating is

$$\begin{aligned} &P_{downbeat}(p|d,c) \\ &\quad = \begin{cases} 1.0, & p \in \text{chord notes of } c, \\ (1-\alpha) \times 1.4/dur \\ + \alpha P_{non,seed}(p|d,c), & p \notin \text{chord notes of } c \end{cases} \end{aligned}$$

$$\begin{aligned} &\text{where} \quad P_{non,seed}(p|d,c) \\ &\quad = \frac{\begin{array}{c}|\text{non-chord tones starting at downbeat} \\ \text{with duration } d \text{ in seed song}|\end{array}}{\begin{array}{c}|\text{notes starting at downbeat with duration} \\ d \text{ in seed song}|\end{array}} \end{aligned} \tag{6}$$

4.4.2. Rhythm style rating functions

We now consider probability distributions regarding rhythm and note duration.

Note duration frequency. Frequency of note duration (length) d appearing in both the seed song statistics and general popular music statistics,

$$\begin{aligned} &P_{durfreq}(d) = \alpha P_{durfreq,seed}(d) \\ &\quad + (1-\alpha)P_{durfreq,general}(d|tempo), \\ &\text{where} \quad P_{durfreq,seed}(d) \\ &= \frac{|\text{notes with duration } d \text{ in seed song}|}{|\text{notes in seed song}|} \end{aligned} \tag{7}$$

Note duration transition. Probability of a note duration given previous consecutive note duration d in both the seed song statistics and general popular music statistics,

$$\begin{aligned} &P_{durtrans}(d_1|d_2) = \alpha P_{durtrans,seed}(d_1|d_2) \\ &\quad + (1-\alpha)P_{durtrans,general}(d_1|d_2), \\ &\text{where} \quad P_{durtrans,seed}(d_1|d_2) \\ &= \frac{|\text{note transitions with duration} d_2 \text{ to } d_1 \text{ in seed song}|}{|\text{note transitions in seed song}|} \end{aligned} \tag{8}$$

Rest note duration. Rest notes are more common to see in the last bar of a phrase/section. For rest notes in the middle of a phrase, we use both rest note duration statistics from the seed song and music theory rules to estimate a probability.

$$\begin{aligned} &P_{durrest}(d|p = rest) \\ &= \begin{cases} 1.0, & \text{section end} \\ \alpha P_{durrest,seed}(d|p = rest) & \text{otherwise} \\ + (1-\alpha) \times 1/d, & \end{cases} \\ &\text{where} \quad P_{durrest,seed}(d|p = rest) \\ &= \frac{|\text{rest note with duration } d \text{ in seed song}|}{|\text{rest note in seed song}|} \end{aligned} \tag{9}$$

Note onset position and duration. Originally, we used a general rule that only prevents syncopation at the 16th note level. For imitation, we want to capture more detail from the seed song, so we estimate the probability of a note duration given each note onset position in 16ths within a bar ($onset_p \in [0 \ldots 15]$). Since the seed song has relatively few notes, we use a form of smoothing. For example, position 4 relative to half notes is equivalent (modulo 8) to position 12, relative to quarter notes (modulo 4) is equivalent to 0, 8 and 12, and relative to eighths (modulo 2) is equivalent to 0, 2, 6, 8, 10, 12 and 14. We use a weighted sum of the statistics for all these positions.

$$\begin{aligned} &P_{pos}(d|onset_p) \\ &\quad = \alpha \left[\sum_{i=1}^{4} \frac{2^i}{16} \sum_{onset_p \equiv j \pmod{2^i}} P_{pos,seed}(d|j) \right] \\ &\qquad + (1-\alpha)P_{pos,general}(d|onset_p), \\ &P_{pos,general}(d|onset_p) \\ &\quad = \begin{cases} 0, & onset_p \equiv 1 \pmod 2 \;\&\; d \equiv 0 \pmod 2 \\ 1, & \text{otherwise} \end{cases}. \end{aligned} \tag{10}$$

4.4.3. Pitch and rhythm dependent rating functions

Harmonic and note duration. Notes longer than quarters are more likely to be chord tones. Notes shorter than quarters are more likely to be non-chord tones. If we use P_{har}, based on song statistics, to capture the notion of chord tones, the following rather obscure equation captures this heuristic:

$$P(p, d|c) = ((P_{har}(p|c) - 0.5) * \log_2(d/4) + 1.0)/2.0. \tag{11}$$

Pitch interval and note duration. Larger intervals occur after longer note durations. We combine a probability estimated from a melody collection with seed song statistics.

$$\begin{aligned} P_{di}(p_2 - p_1|d_1) &= P_{di}(\Delta p|d_1) \\ &= \alpha P_{di,seed}(\Delta p|d_1) \\ &\quad + (1-\alpha)P_{di,general}(\Delta p|d_1), \text{where} \\ P_{di,seed}(\Delta p|d_1) &= \frac{\begin{array}{c}|\text{intervals } \Delta p \text{ with first note} \\ \text{duration } d_1 \text{ in seed song}|\end{array}}{\begin{array}{c}|\text{intervals with first note} \\ \text{duration } d_1 \text{ in seed song}|\end{array}}. \end{aligned} \tag{12}$$

Notes spanning chord changes. Usually, new notes begin at chord changes. We do not allow a note to span two chords unless this occurs in the seed song.

Last bar and the tonic. Sections ending with a strong perfect authentic cadence almost always end on the tonic (i.e. the pitch C) in popular music. The ending note is more likely to have longer duration than shorter notes too. This rule-based constraint overrides probabilistic sampling in our music generation model.

4.4.4. Long-term melody structure

Melody contour. As shown in Figure 7, melody contour (melody shape, configuration, outline) describes the

Figure 7. Grey line represents the melody contour.

general shape of the melody and proves to be very useful for identifying the melody style (Adams, 1976). In order to imitate the melody contour in the seed song, we calculate a *melody pitch contour similarity*, which is based on a Dynamic Time Warping (DTW) (Berndt & Clifford, 1994) algorithm. First, we transform two melody note sequences by taking the pitch or rest at each 16th note. For example, a note with pitch p and duration 4 in the original melody will become 4 consecutive copies of p in the new sequence. We compute the edit distance using a substitution cost based on similar absolute pitch difference and similar melodic direction. We normalise the DTW distance and subtract from 1 to get a similarity.

$$\begin{aligned} cost_{sub}(p_1, p_2) &= w_1 \cdot dist_{pit}(p_1, p_2) \\ &\quad + w_2 \cdot dist_{dir}(prev_1 - p_1, prev_2 - p_2), \\ &\text{where} \quad w_1 = 1.0 \quad \text{and} \quad w_2 = 2.0 \text{ are weights} \\ &\text{and} \quad prev_i \text{ is the pitch before } p_i. \end{aligned} \tag{13}$$

$$dist_{pit}(p_1, p_2) = \begin{cases} 12, & p_1 \text{ or } p_2 \text{ is a rest,} \\ |p_1 - p_2|, & \text{otherwise.} \end{cases} \tag{14}$$

$$dist_{dir}(\Delta_1, \Delta_2) = \begin{cases} 0, & \Delta_1 = \Delta_2 \text{ or } \Delta_1 \cdot \Delta_2 > 0 \\ 2, & \Delta_1 \cdot \Delta_2 < 0 \\ & \text{or } \Delta_1 \text{ or } \Delta_2 \text{ is a rest} \\ 1, & \text{otherwise.} \end{cases} \tag{15}$$

The insertion cost for adding a pitch or rest p_X in sequence X compared to sequence Y is defined as:

$$\begin{aligned} cost_{ins}(p_X, Y) &= \begin{cases} |p_X - \overline{p_Y}|, & p_X \text{ is a pitch} \\ 10, & p_X \text{ is a rest,} \end{cases} \\ &\text{where} \quad \overline{p_Y} \text{ is the average pitch of } Y. \end{aligned} \tag{16}$$

The DTW calculation for the distance between sequences X_i and Y_i is

$$\begin{aligned} dist(X, Y) &= dtw(|X|, |Y|), \text{ where} \\ dtw(i, j) &= \min(dtw(i-1, j-1) + cost_{sub}(X_i, Y_j), \\ &\qquad dtw(i-1, j) + cost_{ins}(X_i, Y), \\ &\qquad dtw(i, j-1) + cost_{ins}(Y_i, X)). \end{aligned} \tag{17}$$

We are generating a new melody left-to-right, so matching a few notes to the entire seed melody contour might incur an extra cost proportional to the difference in lengths. We avoid this problem by matching only the prefix of the seed contour that gives the lowest DTW distance.

Finally, to get from the DTW distance to a similarity, we divide the DTW distance by the distance to a 'flat' melody consisting only of the average seed melody pitch, giving a value from 0 (perfect match) to 1 or more (very poor match). We subtract this from 1 and limit the lowest value to zero to get a similarity:

$$\begin{aligned} P_{sim}(X, Y) &= \max\left(1 - \frac{a}{b}, 0\right), \text{ where} \\ a = dist(X, Y), &\quad b = \text{'poor match' distance.} \end{aligned} \tag{18}$$

Rhythm onsets distribution similarity. Apart from the rhythm style ratings, we also compare the rhythm onsets distribution of the generated imitation to the original seed. The *rhythm similarity metric* is based on matching note onset times. Onsets times are quantised to 16th notes. Given two melodic sequences we define $similarity_{rhythm}$ to be the proportion of 16th note offsets where both sequences or neither sequence contains an onset (i.e. *accuracy*). With sets of onset times A and B (with times measured from the sequence beginnings), and sets of non-onset times A^C and B^C, we define rhythm similarity as

$$similarity_{rhythm} = \frac{|A \cap B| + |A^C \cap B^C|}{|A| + |B| + |A^C| + |B^C|}. \tag{19}$$

As each new duration is considered, we compute this similarity, and if the similarity is below a threshold, we disallow that duration choice.

4.5. Stylistic bass generation

We represent bass style using patterns. First, each bar of bass is represented by a rhythm pattern that denotes onset times. For example, [0____ , 0____ , ______ , 0____] denotes onsets on beats 1, 2 and 4. For each pattern, we form a *Chord Tone Frequency Matrix* with one row for each onset (in this case the rows represent beats 1, 2 and 4). The 4 columns are the probabilities of the bass note being the root, third, or fifth of the chord or a non-chord tone. We also form a *Chord Tone Transition Matrix* for each onset where the columns are the probabilities

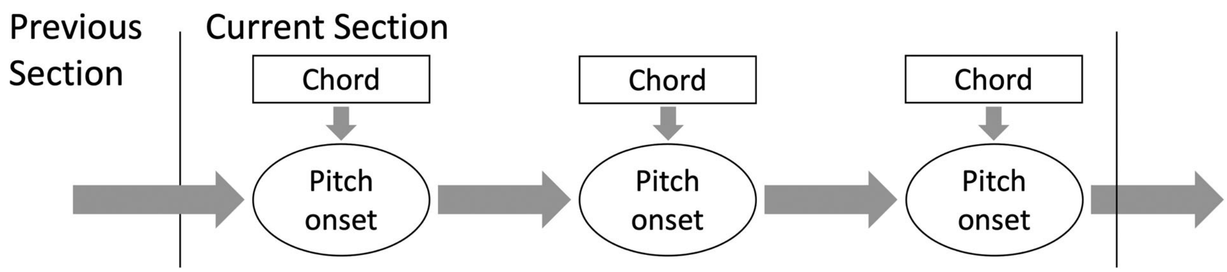

Figure 8. Pattern-based bass generation model using current chord for context and transition matrix for note-to-note probabilities.

of the *next* bass note being the root, third, fifth or non-chord tone. We use the most frequent rhythm pattern in the seed song section together with the two chord tone matrices to generate bass.

Pattern-based bass generation. We generate the stylistic bass section-by-section. The output bass rhythm copies the rhythms of the first and last bars of the seed song, and every other bar uses the most frequent bass rhythm pattern. Bass note selection follows the *Chord Tone Frequency* and *Chord Tone Transition* matrices. These are used to estimate the probability of **p**, a vector representing each bass note as root, third, or fifth. We use the Viterbi algorithm to find the **p** with the maximum likelihood according to this model (Equation (20) and Figure 8).

This is a simple model of bass lines, but it does capture general stylistic characteristics such as always playing the root, alternating root and fifth, or playing arpeggios. It also uses the most common rhythmic pattern, which is a very important aspect of the bass style.

$$\mathbf{p} = argmax_{\mathbf{p}} \prod_i [P(p_i)P(p_i|p_{i-1})],$$
$$p_i = \text{root, third, fifth, other} \tag{20}$$

5. Evaluation and results

We conducted both objective and subjective evaluations of our stylistic music generation system. We collected 13 MIDI songs and manually transformed them to the representation described in Section 3. Three of them are used in the training stage for parameter tuning. The other 10 songs, five Western pop songs and five Chinese pop songs, are used as seed songs for evaluation. These test songs are quite popular and recognisable. One application of our system is to imitate favourite pop songs, so we wanted to imitate songs that are already popular and familiar. Our results show that our stylistic music generation system is able to create music with both high musicality and similarity to the seed song. We further explored the factor of familiarity in music generation.

5.1. Objective evaluation

Objective evaluation of music is not well-understood. In fact, if we had a perfect evaluation function, music generation would become a problem of search and optimisation.

Similarity measures are also quite limited. An exact copy of a melody is the most similar one, but our goal is to create *stylistically* similar music *without* making a direct copy. Objectively, we measure success by showing that our many measures of similarity are satisfied. This can be compared to music-theory-based objective functions proposed by Yang and Lerch (2020). Recall that we developed a set of probability functions to model seed songs, including Equations (3)–(19). We take the product of these probabilities as a measure of similarity for each note and the average probability over all notes as a measure of overall similarity. It should be the case that this similarity measure is higher for imitations than non-imitations. We use a paired t-test to compare similarity of imitations to two types of *non-imitations*. One type is formed by imitating a different song. The other type is formed by generating music with only the general probabilities such as $P_{freq,general}$ without any probability based on the seed song such as $P - freq, seed$. In other words, we set α to 0, giving all the weight to general probabilities. With either type of non-imitation, the estimated probability of stylistic imitations is significantly higher (p-value < 0.001).

Chord sequences use the same paradigm of blending a general probability function with a seed-specific probability function. Here again, we can use the seed-specific probability function to measure similarity to the seed song. As with melody, we found the estimated probability of stylistic chord progression imitations significantly higher (p-value < 0.001) than that of non-imitations (both imitations of other seed songs and chord progressions based only on $P_{general}$).

Since our objective measure of similarity is the *same* probability function used to generate similar songs, the result is not surprising. Nevertheless, it is important to confirm that after all the complex interactions of these

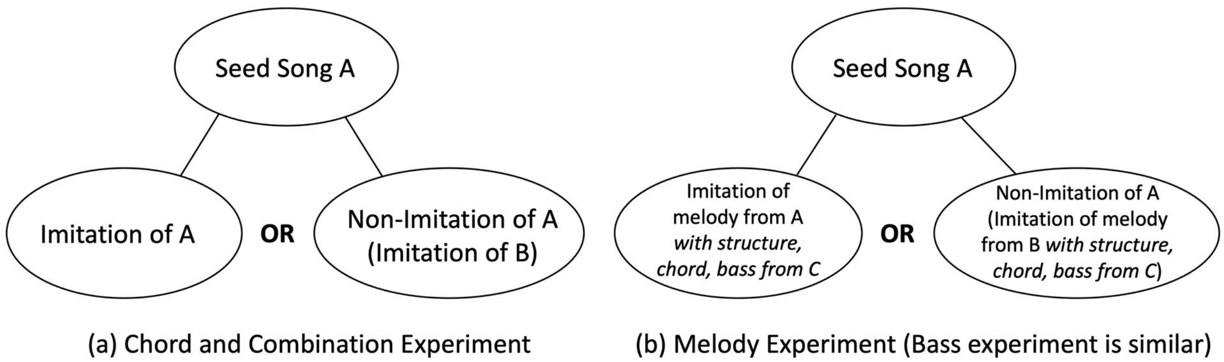

Figure 9. Settings for each pair of examples in the experiment.

rules and random sampling, our goal of biasing composition to incorporate features of seed songs is successful.

At the same time, we do not want similarity to come at the cost of overall rating. We should at least expect the generated songs to have approximately the same probability as their seed songs in terms of our rating functions. A paired t-test shows that there is no significant difference in estimated probabilities (objective rating functions) between the original seed songs and the generated imitations (p-value $= 0.211$). Moreover, we find that 57% of the generated imitations have higher probability than their seed songs. Thus, our approach can produce music that is similar to human-composed music in terms of our objective rating functions. This is consistent with subjective evaluations discussed in the next section.

5.2. Subjective evaluation

Our objective evaluation shows that our methods are behaving as intended from a statistical point of view, but our main objectives relate to creating enjoyable music. Direct evaluations of our claims require human rating. The following sections describe the songs we constructed, the raters and their ratings, statistical summaries of the ratings, and the implications of these ratings.

5.2.1. Evaluation design

We conducted human listening evaluations for the melody, chord and bass modules in our system and for the combination of all three. Listeners first answered questions about their music background, for example, their music practice experience, music genre preference, age and gender. Then, the listeners answered four parts of the evaluation (melody, chord, bass and combination) in a random order. In each part, listeners compared and rated 10 example pairs. Within each example pair, there are two music fragments to listen to, each of them around 30 seconds in length. As shown in Figure 9, one fragment is the seed song. The other is either a stylistic imitation based on the seed song (called an *imitation*) or an imitation of some other seed song (called a *non-imitation*). We asked users to 1) indicate if they have heard the music fragments before the listening survey; 2) judge the similarity between the two music fragments in each example pair; 3) rate how much do they like the music fragments. Both similarity and preference score ranges use a Likert scale labelled with integers from 1 to 5. Every user will randomly get either an imitation or a non-imitation version for each seed song (seed song and the generated song are displayed in random order), and for each part, there are five imitations and five non-imitations. The 10 seed songs are also presented in random order for each part. Users must listen to the full length of the songs, otherwise their ratings are not counted.

As shown in Figure 9(b), for the melody study, we want the similarity to the seed song to depend only on whether the melody imitates the melody of the seed song (A), or another song (B). To remove any coordination with chords, bass and structure, we derive these non-melodic elements by imitating a third song (C). To illustrate the problem this solves, imagine if the imitation uses the same chords as the seed song. The listener might rate the melody as similar merely because of a similar chord progression. In addition, the combination of melodic contour and chords might be enough to substantially reproduce the seed song melody. We remove this information by taking chords from another song. For the bass similarity study we take a similar approach, imitating the melody, chords and structure from a third song (C) in both the *imitation* and *non-imitation* cases. In order to help non-musicians identify melody and bass, we give examples of what is a melody or bass line before the evaluation. For the chord similarity study (Figure 9(a)), we present only simple block chords from the songs with no melody or bass. For the combination study, we imitate all parts including melody, chord, and bass.

We use MIDI and software synthesisers to produce audio versions of the songs after adjusting several parameters to minimise the effect of timbre, pitch and

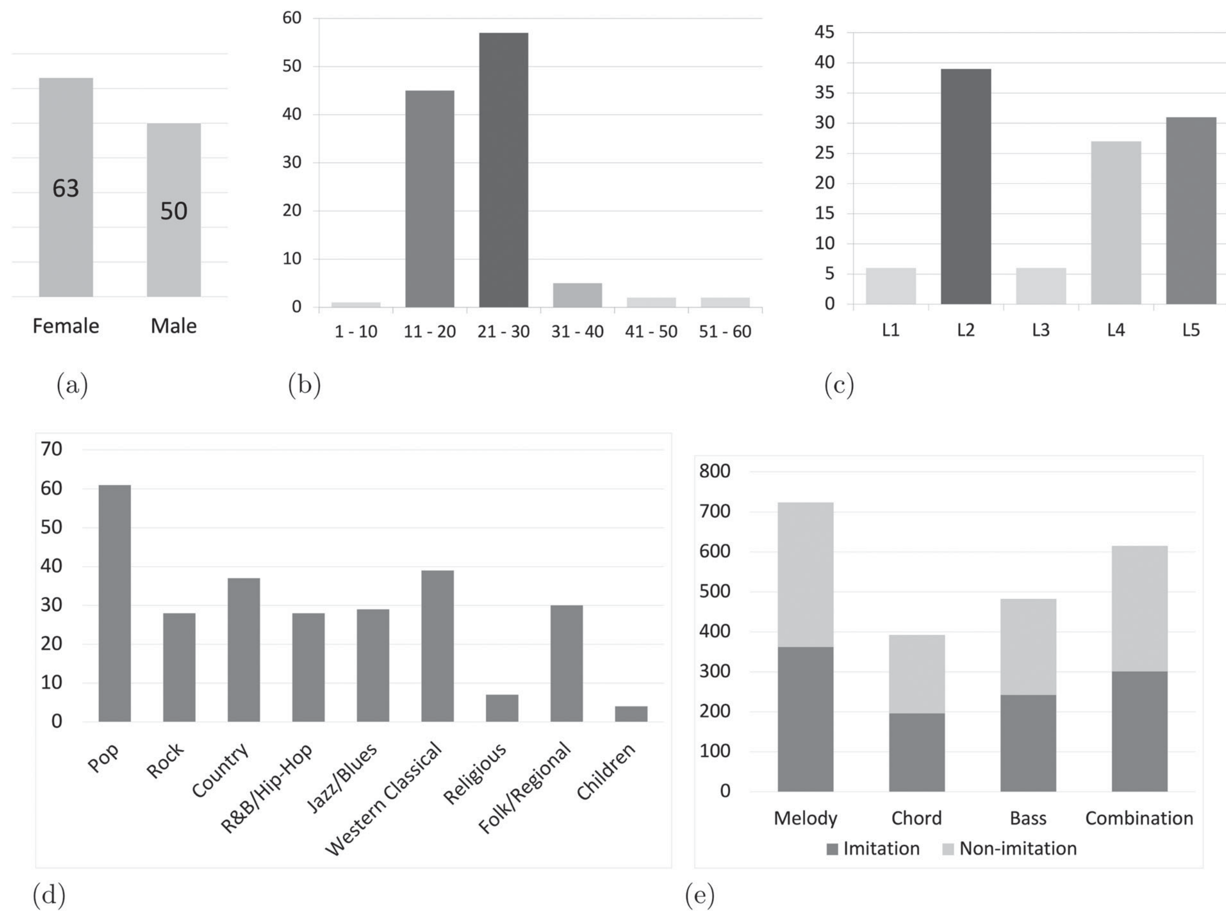

Figure 10. Demographics Distribution. (a) Gender. (b) Age. *X*-axis are different age groups. (c) Music Level. L5 means expert. (d) Music Preference. Users can select multiple genres. (e) Number of ratings collected for each evaluation part.

tempo differences on similarity ratings. Within any pair of song fragments presented to raters, the synthesiser settings are identical. We also shift octaves to make the average pitch as close as possible and use the same tempo within each example pair. Here we use the original production and song settings of the seed song for each pair.

5.2.2. Ratings, results and discussion

We collected 154 evaluation surveys. We removed surveys that are not reliable based on validation and confidence tests. For example, some users gave different scores for the same seed songs in different parts; some users made inconsistent claims about whether they have heard the same song; some users even indicated that they have heard most of our system generated songs before. All these surveys are considered too unreliable to use. After removing these surveys, we are left with 2214 rating pairs from 113 listeners. The demographics information about the raters that provided ratings that were actually used are shown in Figure 10.

Observation 5.1: Our system can generate good music overall.

A problem with measuring preference is that familiar music is generally preferred over unfamiliar music (Pereira et al., 2011; Ward et al., 2014). Therefore, before the evaluations, we asked the users to indicate whether they have heard the seed songs before. In our study, based on an unpaired t-test, except for the chord progression part, listeners prefer familiar seed songs more than unfamiliar seed songs (p-value < 0.001), as shown in Figure 11. There are many possible reasons that familiar chord progressions are not preferred: Listeners found chord progressions in isolation difficult to evaluate, chord progressions are slower and may be less memorable or recognisable than melody, and familiar chords without familiar melody might create unfulfilled expectations.

Of course *all* of our generated music is new and unfamiliar to listeners. Furthermore, we learned that some users consciously lowered scores for computer-generated songs when they recognised the seed song. To level the playing field, we consider only rating pairs where seed songs are not familiar. Using unpaired t-tests, we find that for all evaluation parts, if the seed song is unfamiliar, there is no significant difference between preference for

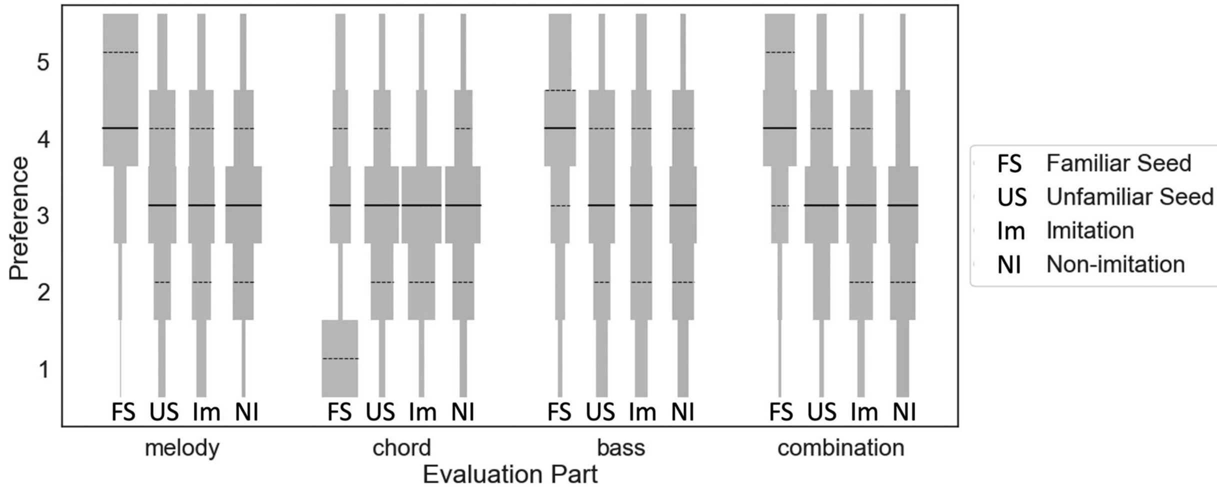

Figure 11. Preference ratings for the familiar and unfamiliar seed songs, imitations and non-imitations. Ratings of machine-generated imitations and non-imitations, when paired with familiar seed songs, were rated lower and are excluded here.

the seed song and preference for either imitations or non-imitations (all p-value > 0.05, average p-value $= 0.298$). Moreover, if the seed song is unfamiliar, 58% of the imitations have the same preference ratings, and 17% of the imitations have higher preference ratings, compared to their paired seed songs. Thus, the overall quality of our generated songs are usually as good, and sometimes better, than their paired seed song unless the seed song is already familiar. Non-imitations are imitations of some other (also unfamiliar) seed song, which explains why the preferences of imitations and non-imitations have no measurable differences.

Observation 5.2: The imitation techniques in our system are effective: Imitations are judged to be similar to the seed songs.

T-tests between the similarity ratings for imitations and non-imitations show that imitations are judged significantly more similar (p-value < 0.001) to their paired seed songs than non-imitations, except for chord evaluation (Figure 12). Furthermore, when users are not familiar with the seed song, the similarity of imitations for melody and combination evaluation is judged significantly higher than when the seed song is familiar (but only for melody and all features combined). Thus, it appears that familiarity affects the perception of similarity. Perhaps when listeners are more familiar with the seed song, they form strong expectations and are therefore more attentive to differences between the imitation and seed song.

Observation 5.3: The more preferable the seed song is, the more preferable the imitation will be.

Table 1. Pearson correlation coefficient test for preferences of seed songs and its imitations/non-imitations.

Correlation	Melody	Chord	Bass	Combination
Seed and imitation	0.42	0.75	0.63	0.28
Seed and non-imitation	0.22	0.67	0.33	0.13
p-value of comparison	< 0.001	> 0.05	< 0.001	< 0.001

One important assumption and motivation for this research is that if we imitate the styles of the seed songs that people prefer, then they will also like the imitations. In other words, the more listeners like the seed song, the more they should like our imitation. To test this, consider Figure 9(a). Users rate songs in pairs of seed song and generated song, with or without imitation. We express ratings as Z-scores in order to normalise the bias and standard deviations of different raters. Let A represent the preference rating for song A, and let A' represent the preference rating for an imitation of A, while B' represents the preference rating for a non-imitation of A. We then compute the correlation of A with B' as a control condition, and A with A' as the experimental condition. As shown in Table 1, for the control condition, the correlation is significantly positive at 0.13. One possible explanation for this is that we always use the same sounds to synthesise the seed song and generated song to enable fair comparisons. The resulting shared timbral and production qualities are likely to influence ratings. For the experimental condition, the correlation is 0.28, which is significantly higher than the control ($p < 0.001$). Therefore, we conclude that the more listeners like seed song A, the more they like imitation A', even after controlling for production value preferences of timbre, tempo and octave.

Feedback from listeners. We also received feedback and comments from the users. For example, some users said

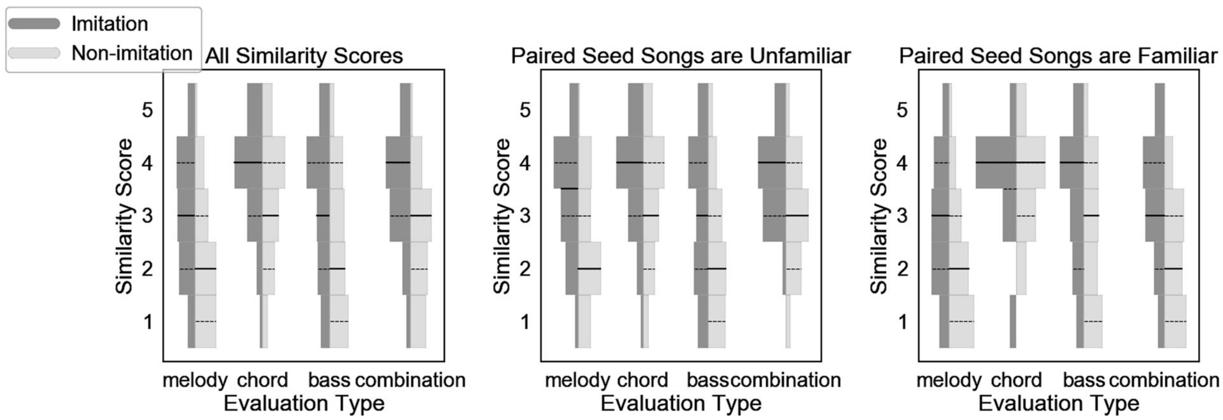

Figure 12. Similarity ratings between seed songs and generated songs, showing differences between imitations and non-imitations. The second and third plots separate the data to consider only cases where the seed songs are unfamiliar and familiar, respectively.

that when they did the later part the survey, they can guess there is one computer-generated song and one human-composed song in each paired example, because they are familiar with some of the human-composed seed songs. Once they figured out there is one human-composed song that they have heard before, they consciously rated the other generated song lower in the paired example. Some also told us that the chord progression is too hard for non-musical experts to evaluate, and even hard to tell the difference for professional musicians, which explains why the results for chord progression alone are not consistent with the other parts. For bass evaluation, even though we gave examples to users to explain what is a bass line, some non-musician users still said the melody and production sometimes distracted their attention from the bass when rating similarity and preference. Also, we tried to control for instrumentation and timbral preferences by using the same production rendering within each paired example, but the production might be more suitable for one genre or nationality of song, and might lead to some bias in the rating. There are many things to consider when conducting listening tests for computer generated music, but so far there is no standard methodology. The practices we used here including confidence tests to filter out invalid evaluations, eliminating other irrelevant factors in the songs, recording whether users have listened to the full length of the songs, listening environment control, etc., may be useful in other studies.

6. Conclusions

We have described techniques to automatically imitate melody, harmony and bass lines for pop songs with a logical and hierarchical music structure. Our original motivation was to enable the creation of therapeutic music where repeated listening might call for some variety, but where therapists might not have the time, resources or permission to use existing popular music. One can imagine many other 'music on demand' applications such as background music for slides and videos or interactive music for games.

We hoped that by imitating favourite songs, we could make music more personalised and enjoyable than 'generic' music based on norms but without any particular style. To demonstrate this, we first needed to meet two requirements: (1) generate desirable, likeable music, (2) generate music that is similar to *seed* songs. Then, we showed that (3) listeners prefer our imitations of songs they like over imitations of songs they like less. This result shows great promise that music can be generated for a wide range of listeners and preferences using favourite songs and imitation to improve the results.

From experience, we believe that the melodic contour and rhythmic onset similarity play the most important roles in generating similar songs, at least within the probabilistic framework that we introduced. However, even drastic changes, such as disabling a rule entirely, do not generally cause obvious changes in the output. This is at least partly due to the probabilistic approach and the fact that rules are not independent. For example, if the pitch interval frequency rule (Equation (5)) is ignored, the pitch interval with harmony, pitch interval and note duration (Equation (12)), and melody contour (Equation (17)) all reinforce smaller intervals in other ways. Keep in mind also that these rules are applied note-by-note and only contribute to overall probabilities. In many cases, random sampling leads to rule 'violations' in the sense that the results are not very imitative. In our experience, these 'violations' are often advantageous because they prevent the music from being too close to the original, avoiding the 'uncanny valley' effect and often resulting in distinctive and creative surprises. When reinforced through repetition, these 'surprises' can give character to otherwise normative melodies. While we have no formal model of surprise and novelty, it seems to be helpful in small doses as suggested in Figure 1.

Further study and tuning of our rules would be interesting but would require extensive human judgements or an objective evaluation function. It should also be mentioned that differences in timbre and production qualities are very important to perception, genre, and musical taste. Timbre, instrumentation and other production qualities fall outside our study of rhythm, bass, melody and harmony, so we carefully controlled for their effect in our studies. Certainly, these factors should be exploited if the goal is to enhance music preference through imitation.

In addition to our encouraging results, we are just beginning to explore the possibilities of combining different songs and genre, for example, combining the rhythm, bass, melody and harmony of as many as four different songs in order to create hybrid songs. With the addition of some editing capabilities to inject human suggestions, this could be an enjoyable creative activity, even for novices with little musical training.

We are also working to combine some of these ideas with deep learning approaches. Much of our success comes from making good choices for the next note based on probabilistic models informed by music theory and simple statistics. Perhaps more sophisticated sequence learning could make even better choices without losing the benefits we gain by incorporating knowledge of music structure and harmony in our process. We also believe that many ideas in this paper can inform future systems, serve as a baseline for comparison and evaluation, and offer insights into music perception and cognition.

Acknowledgments

We would like to thank Larry Wasserman and Roy Maxion for their ideas and guidance on statistical analysis. Open Access funding provided by the Qatar National Library.

Disclosure statement

No potential conflict of interest was reported by the author(s).

References

Adams, C. R. (1976). Melodic contour typology. *Ethnomusicology*, *20*(2), 179–215. https://doi.org/10.2307/851015

Berlyne, D. E. (1971). *Aesthetics and psychobiology*. Appleton-Century-Crofts.

Berndt, D. J., & Clifford, J. (1994). Using dynamic time warping to find patterns in time series. In *Proceedings of the 3rd international conference on knowledge discovery and data mining* (pp. 359–370). AAAI Press.

Briot, J. P., Hadjeres, G., & Pachet, F. (2019). *Deep learning techniques for music generation*. Springer.

Chmiel, A., & Schubert, E. (2017). Back to the inverted-U for music preference: A review of the literature. *Psychology of Music*, *45*(6), 886–909. https://doi.org/10.1177/0305735617697507

Conklin, D., & Witten, I. H. (1995). Multiple viewpoint systems for music prediction. *Journal of New Music Research*, *24*(1), 51–73. https://doi.org/10.1080/09298219508570672

Cope, D. (1991). *Computers and musical style* (Vol. 6). Oxford University Press.

Cope, D. (2000). *The algorithmic composer* (Vol. 16). AR Editions, Inc.

Cope, D. (2005). *Computer models of musical creativity*. MIT Press.

Dai, S., & Xia, G. G. (2018). Music style transfer: A position paper. In *Proceedings of 6th international workshop on musical metacreation*. https://doi.org/10.48550/arXiv.2010.07518.

Dai, S., Zhang, H., & Dannenberg, R. B. (2020). Automatic analysis and influence of hierarchical structure on melody, rhythm and harmony in popular music. In *Proceedings of the 2020 joint conference on AI music creativity (CSMC-MuMe 2020)*. https://doi.org/10.48550/arXiv.2010.07518.

Dannenberg, R. B. (1993). Music representation issues, techniques, and systems. *Computer Music Journal*, *17*(3), 20–30. https://doi.org/10.2307/3680940

Elowsson, A., & Friberg, A. (2012). Algorithmic composition of popular music. In *Proceedings of the 12th international conference on music perception and cognition and the 8th triennial conference of the European society for the cognitive sciences of music* (pp. 276–285). http://urn.kb.se/resolve?urn = urn:nbn:se:kth:diva-109400.

Hadjeres, G., Pachet, F., & Nielsen, F. (2017). Deepbach: A steerable model for Bach chorales generation. In *Proceedings of the 34th international conference on machine learning-volume 70* (pp. 1362–1371).

Hakimi, S. H., Bhonker, N., & El-Yaniv, R. (2020). Bebopnet: Deep neural models for personalized jazz improvisations. In *Proceedings of the 21st international society for music information retrieval conference, ISMIR 2020*.

Hiller, L., Ames, C., & Franki, R. (1985). Automated composition: An installation at the 1985 international exposition in Tsukuba, Japan. *Perspectives of New Music*, *23*(2), 196–215. https://doi.org/10.2307/832731

Huang, C. Z. A., Cooijmans, T., Roberts, A., Courville, A., & Eck, D. (2017). Counterpoint by convolution. In *Proceedings of the 18th international society for music information retrieval conference, ISMIR*.

Huang, C. Z. A., Vaswani, A., Uszkoreit, J., Simon, I., Hawthorne, C., Shazeer, N., Dai, A., Hoffman, M., Dinculescu, M., & Eck, D. (2019). Music transformer: Generating music with long-term structure. In ICLR. https://www.semanticscholar.org/paper/Music-Transformer%3A-Generating-Music-with-Long-Term-Huang-Vaswani/fb507ada871d1e8c29e376dbf7b7879689aa89f9

Jiang, Z., & Dannenberg, R. (2019). Melody identification in standard MIDI files. In *Proceedings of the 16th sound & music computing conference* (pp. 65–71).

Kawai, L., Esling, P., & Harada, T. (2020). Attributes-aware deep music transformation. In *Proceedings of the 21st international society for music information retrieval conference, ISMIR* 2020.

Liang, F. (2016). Bachbot: Automatic composition in the style of Bach chorales. *University of Cambridge*, *8*, 19–48.

Lo, M., & Lucas, S. M. (2006). Evolving musical sequences with n-gram based trainable fitness functions. In *2006*

IEEE international conference on evolutionary computation (pp. 601–608).
Mori, M. (1970). The uncanny valley. *Energy*, *7*(4), 33–35.
Müllensiefen, D., Lewis, D., Rhodes, C., & Wiggins, G. (2007). Treating inherent ambiguity in ground truth data: Evaluating a chord labelling algorithm. In *8th international conference on music information retrieval (ISMIR)*.
Payne, C. (2019). *MuseNet*. https://openai.com/blog/musenet.
Pearce, M. T. (2005). *The construction and evaluation of statistical models of melodic structure in music perception and composition* [Unpublished doctoral dissertation]. Department of Computer Science, City University of London, UK.
Pereira, C. S., Teixeira, J., Figueiredo, P., Xavier, J., Castro, S. L., & Brattico, E (2011). Music and emotions in the brain: Familiarity matters. *PLoS ONE*, *6*(11), https://doi.org/10.1371/journal.pone.0027241
Roberts, A., Engel, J., Raffel, C., Hawthorne, C., & Eck, D. (2018). A hierarchical latent vector model for learning long-term structure in music. In *Proceedings of the 35th international conference on machine learning*.
Sluckin, W., Hargreaves, D., & Colman, A. (1983). Novelty and human aesthetic preferences. In J. Archer & L. L Birke (Eds.), *Exploration in animals and humans* (pp. 245–269). Van Nostrand Reinhold.
Thaut, M. H., McIntosh, G. C., Rice, R. R., Miller, R. A., Rathbun, J., & Brault, J. (1996). Rhythmic auditory stimulation in gait training for Parkinson's disease patients. *Movement Disorders: Official Journal of the Movement Disorder Society*, *11*(2), 193–200. https://doi.org/10.1002/(ISSN)1531-8257
Thornton, C. (2009). Hierarchical markov modeling for generative music. In *International computer music conference*.
Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems* (pp. 5998–6008).
Ward, M. K., Goodman, J. K., & Irwin, J. R. (2014). The same old song: The power of familiarity in music choice. *Marketing Letters*, *25*(1), 1–11. https://doi.org/10.1007/s11002-013-9238-1
Yang, L. C., & Lerch, A. (2020). On the evaluation of generative models in music. *Neural Computing and Applications*, *32*(9), 4773–4784. https://doi.org/10.1007/s00521-018-3849-7
Yang, R., Wang, D., Wang, Z., Chen, T., Jiang, J., & Xia, G. (2019). Deep music analogy via latent representation disentanglement. In *20th international society for music information retrieval conference*.
Yu, L., Zhang, W., Wang, J., & Yu, Y. (2017). Seqgan: Sequence generative adversarial nets with policy gradient. In *Thirty-first AAAI conference on artificial intelligence*.

Appendix

A.1 User study questionnaire

Instructions before starting the survey:

- The survey will take about 50 minutes (5 parts in total). Remember to save your answers before you quit. You can resume and continue the survey anytime you want.
- We highly recommend you do this survey on your computer and use your headphones.
- If you meet any problem during the test, please contact us via email.
- Remember to leave comments after the survey.

Part I. Music Background

- Music listening and Practice (select one from the following)
  - I am an expert, e.g. more than 5 years of vocal or instrumental training and practice.
  - I studied an instrument or voice 1 to 5 years.
  - I have less than one year study but listen at least 15 hours per week.
  - I like music and listen 1 to 15 hours per week.
  - I listen to music less than 1 hour per week.
- What genre of music do you prefer? (multiple selection)
  - Pop, Rock, Country, R&B/Hip-Hop, Jazz/Blues, Western Classical, Religious, Folk/Regional, Children
- What is your age? (select one from the following)
  - 5–10, 11–20, 21–30, 31–40, 41–50, 51–60, > 60
- What is your gender? (type in)

Part II. Melody Evaluation (The order of Part II, III, IV is random.)

In this section, we want you to judge the similarity between **melodies** in two songs. Examples are fragments extracted from full songs. Some of them are generated by AI algorithm, and some are written by human composers. Same instructions for all examples:

Listen to *Song A* and answer the following two questions, then listen to *Song B* and answer the following three questions. For rating questions, try to use the **full range** from 1 to 5 stars. You must complete listening to the **full length of the song** before answering its questions. You may go back and listen to the two songs as many times as you wish.

If you are confused what is melody, listen to the example below:

This is a song (play a song). This is its melody (play its melody).

There are 10 examples, each example is described as:

Example *n*

- play *Song A*
- Have you listened to Song A before this survey? (yes or no)
- How much do you like the melody of Song A? (rate 1 to 5)
- play *Song B*
- Have you listened to Song B before this survey? (yes or no)
- How much do you like the melody of Song B? (rate 1 to 5)
- How similar is the melody of Song A compared to Song B? (rate 1 to 5)

Part III. Chord Evaluation

Similar to melody evaluation, change *melody* to *chords* in the instructions and questions.

Part IV. Bass Evaluation

Similar to melody evaluation, change *melody* to *bass line* in the instructions and questions.

Part V. Combination Evaluation

Change the first sentence of instructions to: In this section, we want you to judge the **overall similarity (including melody, chord and bass)** between two songs.

Change the question 'How much do you like the melody/bass lines/chords of Song A/B' to 'How much do you like Song A/B?'