




Multi-mmlg: a novel framework of extracting multiple main melodies from MIDI files

Jing Zhao¹ · David Taniar² · Kiki Adhinugraha³ · Vishnu Monn Baskaran¹ · KokSheik Wong¹ 

Received: 7 August 2022 / Accepted: 24 July 2023 / Published online: 16 August 2023
© The Author(s) 2023

Abstract

As an essential part of music, main melody is the cornerstone of music information retrieval. In the MIR's sub-field of main melody extraction, the mainstream methods assume that the main melody is unique. However, the assumption cannot be established, especially for music with multiple main melodies such as symphony or music with many harmonies. Hence, the conventional methods ignore some main melodies in the music. To solve this problem, we propose a deep learning-based Multiple Main Melodies Generator (Multi-MMLG) framework that can automatically predict potential main melodies from a MIDI file. This framework consists of two stages: (1) main melody classification using a proposed MIDIXLNet model and (2) conditional prediction using a modified MuseBERT model. Experiment results suggest that the proposed MIDIXLNet model increases the accuracy of main melody classification from 89.62 to 97.37%. In addition, this model requires fewer parameters (71.8 million) than the previous state-of-art approaches. We also conduct ablation experiments on the Multi-MMLG framework. In the best-case scenario, predicting meaningful multiple main melodies for the music are achieved.

Keywords Melody extraction · Main melody · MIDI · Multiple melodies

David Taniar, Kiki Adhinugraha, Vishnu Monn Baskaran and KokSheik Wong have contributed equally to this work.

✉ KokSheik Wong
wong.koksheik@monash.edu

Jing Zhao
jing.zhao@monash.edu

David Taniar
david.taniar@monash.edu

Kiki Adhinugraha
k.adhinugraha@latrobe.edu.au

Vishnu Monn Baskaran
vishnu.monm@monash.edu

¹ School of Information Technology, Monash University Malaysia, Bandar Sunway, Malaysia

² Faculty of Information Technology, Monash University, Melbourne, Australia

³ Department of Computer Science and IT, Latrobe University, Melbourne, Australia

1 Introduction

Main melody is essential information of a piece of music. Main melody can be applied in various applications, including music retrieval [1, 2], accompaniment generation [3, 4], melody plagiarism identification [5, 6], cover song recognition [7, 8], and new melody generation [9–11]. Consequently, automatic extraction of the main melody becomes more important [12, 13], and it has captivated the attention of many researchers. Humans possess different cognitive attributes, including auditory perception and deliberate attention [14] which in turn enables us to subjectively identify the main melody through loudness transformation, pitch combination, and so forth. However, computers lack perceptual capacity and subjective judgment. Therefore, it would be arduous for a computing system to automatically extract the main melody [1, 13, 15].

Music is usually recorded in audio and symbolic files [2, 16], which are completely different. On the one hand, the audio files (e.g., WAV) encode the signal information of music, and they are realistic recordings of natural sounds, including noise and perceptual information such as

loudness and intensity. On the other hand, a symbolic file (e.g., MIDI—Musical Instrument Digital Interface) is essentially a sequence of note's *messages*, including velocity, pitch, and duration, track [2, 17, 18]. The symbolic file can better reflect the content of music in comparison to the audio file [16]. Hence, the focus of this work is on extracting melody representations from a symbolic file, namely MIDI.

Conventional methods use the most distinctive feature of MIDI files (track message) to obtain the main melody [12, 16, 19–22]. These methods assume that the main melody notes are located on a single track, and therefore, they directly identify the main melody track using some statistical characteristics of the track of interest. However, such assumptions cannot be made in real-world application because this approach represents an ideal condition, which rarely occurs.

Aside from obtaining the track feature of a MIDI file, data-driven approaches using deep learning techniques could be used to extract the main melody notes. Such techniques include both the classification and prediction approaches. Specifically, classification approaches such as MIDIBERT [24], CNN [13] and LStoM [25] could only output one main melody. In other words, they assume that the main melody of a piece of music is unique by default. Under this assumption, these methods classify all notes as either main melody or accompaniment. These classification methods can give better results than algorithmic methods such as Skyline [15] or methods that filter out accompaniment notes using thresholds of pitch interval [26]. Unlike the classification approaches, prediction approaches are more flexible and could predict multiple melodies, such as MusicFrameworks [10]. Specifically, Dai et al.'s method [10] uses basic melody and other music information to generate new melodies. The basic melody could be understood as the main melody. Although MusicFrameworks [10] can actually predict multiple basic melodies, the predicted results have low similarity with the corresponding music, or in other words, the results are not the main melodies. Hence, there are no existing methods that can successfully extract multiple main melodies.

Although most conventional methods assume the uniqueness of the main melody, this assumption is not reasonable. To be specific, music often contains multiple main melodies that are equally important [27], and hence, it is challenging to determine the single or the essential main melody [19]. For example, in Fig. 1, the choral music in the Bach10 dataset [23] contains at least two melodies, i.e., Melody 1 and Melody 2, where both of them could be used to identify the music. If using the MIDIBERT [24] method that defaults to the main melody uniqueness for extracting the choral music's main melody, the extracted result is highly similar to Melody 1 according to the

experiment results. In this case, if the main melody of other music is similar to choral music's Melody 1, other music and the choral music will be matched using the main melodies. However, the main melody of Music A in Fig. 1 is similar to choral music's Melody 2, so Music A and the choral music could not be matched if the main melody of the choral music is only Melody 1. Consequently, assuming music only has a unique main melody has limitations and contradicts the complexity of the main melody. For this problem, if both Melody 1 and Melody 2 of the choral music could be identified as the main melodies, the matching in Fig. 1 would be successful. This problem is usually found in applications where the main melodies are used for retrieval, such as music retrieval [1, 2], melody plagiarism identification [5, 6], and cover song recognition [7, 8]. Hence, an approach that could identify multiple main melodies within a piece of music is required.

If all main melodies are highly similar and highly relevant to the corresponding music, it will be meaningless to predict multiple main melodies. Conversely, if the extracted melodies are entirely different or irrelevant to the music of interest, then, the prediction has failed because melodies may actually correspond to different music. Therefore, only melodies that are different but remain moderate similarity with respect to the music of interest and viable for identifying the music are the main melodies. As a result, to extract multiple main melodies, we need to address the following difficulties: (1) Finding an approach to predict multiple main melodies and (2) Finding a strategy to control the similarity of the final predicted results to ensure that they are the main melodies.

According to the above analysis, the main melody classification methods assume the uniqueness of main melodies and only output one result, so they cannot accommodate the complexity and diversity of main melodies. The advantage of these methods is the classified main melody with high accuracy. Furthermore, the advantage of prediction approaches is that they can predict multiple melodies, but they are usually used in new melody generation tasks rather than the main melody extraction task because of their uncontrollable predicted results. In conclusion, neither the existing melody classification methods nor the prediction methods can address the multiple main melodies prediction problem independently. However, they all have merits. Hence, we propose a novel framework combining the merits of classification and prediction approaches. We name our framework as Multi-MMLG (Multiple Main Melodies Generator) framework, and Fig. 4 shows the structure of the framework. Note that Multi-MMLG is a 2-stage framework, where

- Stage 1: The classified main melody with high accuracy is a good condition for predicting main melodies.

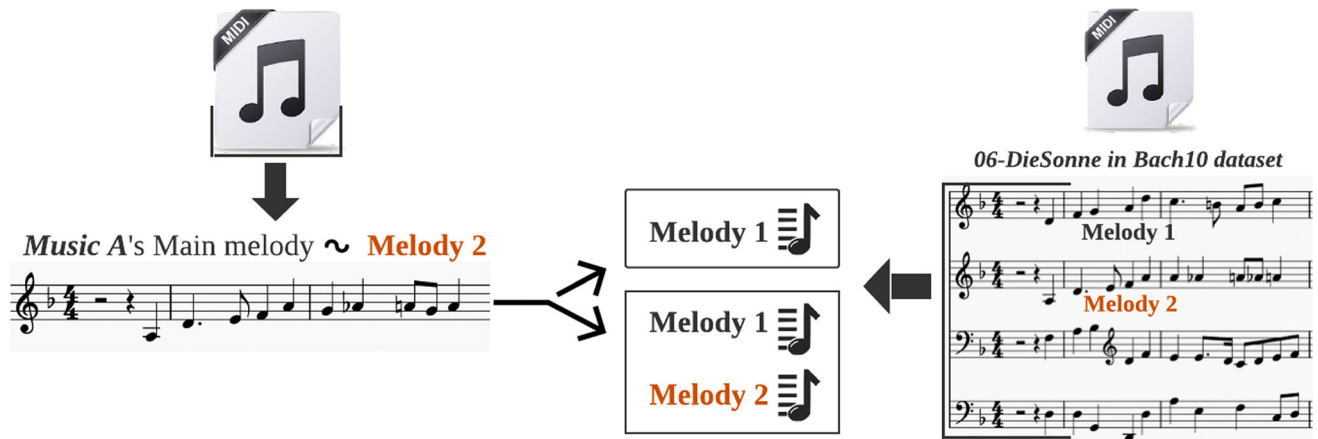


Fig. 1 An example to illustrate a limitation of assuming that the main melody is unique. The Choral music from the Bach10 dataset [23] (06-DieSonne) contains at least two main melodies, namely Melody 1 and 2. If the main melody of other music (e.g., Music A) is similar to

Melody 2, the choral music and Music A should be similar from the human perspective. However, if the only extracted main melody of the choral music is Melody 1, matching these two pieces of music using the main melody may fail

Hence, we set the first stage of Multi-MMLG framework to a note level classification stage thus providing a high-accuracy classified main melody as the next stage's input. To obtain a higher accuracy than using existing methods, we implement a MIDIXLNet model, and we use it in the framework's first stage.

- Stage 2: Since the prediction method is suitable to output multiple melodies, the second stage of the Multi-MMLG framework is a conditional prediction stage. Considering that the predicted melody normally has low similarity with the corresponding music, we use two conditions, Stage 1's classified main melody and a notes' relationship matrix from the original MuseBert approach [28]. These two conditions will be used as the input of a modified MuseBERT model, and they can constrain the predictions to ensure that the predicted melodies are the main melodies. Moreover, because the main melodies should be similar and relevant to the corresponding music at a moderate level, neither too high nor too low, we implement a masking strategy on the predicting conditions to control the predicted results.

The evaluation results of the Multi-MMLG framework are analyzed quantitatively and qualitatively. Our results demonstrate that the framework could extract potential main melodies, and the similarity of the predicted results can be controlled within a reasonable range. In addition, we also conduct ablation experiments and compare our framework with other approaches. The ablation results verify the framework's effectiveness and rationality.

To the best of our knowledge, this is the first work that details and clarifies the definition of multiple main melodies. The contributions of this paper include:

1. Putting forward a new definition for the *main melody*, which is a set of similar but non-identical melodies that can be analyzed by entities (humans and computers) to identify a piece of music. This definition of main melody then serves as the backbone of this work;
2. Implementing a MIDIXLNet model which increases the accuracy of the main melody classification task to 97.37% with relatively lower parameter numbers than existing methods. Meanwhile, we verify the necessity of designing a model specifically for the midi-based main melody classification task;
3. Proposing a two stages Multi-MMLG framework that could predict multiple main melodies. It combines MIDIXLNet and a modified MuseBERT model. Furthermore, the framework avoids the randomness of the prediction melodies by modifying the prediction strategy, notes' representation, and implementing a masking strategy. Ablation experiments demonstrate that the combination of MIDIXLNet and a modified MuseBERT model is optimal.

The structure of the paper is as follows: In Sect. 2, the preliminary knowledge referenced in this article is presented, and this section reviews the conventional methods designed for main melody extraction. Section 3 puts forward the Multi-MMLG framework. Experiment results are presented and discussed in Sect. 4. Finally, Sect. 5 concludes this paper.

2 Preliminary and related work

This section will firstly introduce two types of digital music files: audio and symbolic. Next, some existing works will be discussed, including MIDI-based and audio-based.

Finally, a new main melody definition will be clarified. This definition is the cornerstone of proposing our framework.

2.1 Digital music files

Digital music files can be broadly divided into two categories, namely audio files and symbolic files. In the former case, an audio file encodes the signal information of a piece of music, namely the physical information [29]. The audio files could present the sound of the natural world to the greatest extent, including noise. Unlike the audio file, symbolic files (e.g., MIDI file) directly store the music as a sequence of *messages* rather than the signal information. Hence, they usually exclude noise and can reflect the content of music better than the audio file [16]. In this work, we focus on symbolic file, in particular, the MIDI file. Figure 2 shows a snippet of a MIDI file that describes one *bar/measure* of music score with $\frac{4}{4}$ *time signature*. In a MIDI file, each note is represented as eight messages (viz., eight-tuple), which include Track, Channel, Pitch, Velocity, and so forth [2, 17, 18]. A MIDI file is usually visualized using *piano roll* based on Track, Pitch, and Time messages [30].

In an ideally structured MIDI file, the notes are grouped by instruments or main/non-main melody, and each group is located in separate MIDI tracks [15, 22]. However, a common scenario is when the notes are located on the same MIDI track. Hence, extracting the main melody from such MIDI files is more complex than an ideally structured file, to which this paper focuses on this complex scenario.

2.2 Related works

The existing methods of main melody extraction are mainly based on audio files and MIDI files. For the purposes of this work, we focus on MIDI-based methods, which can be broadly categorized into 2 groups, namely: the track/bar-level methods and the note level methods. Subsequently, we briefly discuss the audio-based methods.

2.2.1 MIDI-based methods

Track and measure level methods The early MIDI-based main melody extraction methods are essentially extracting the MIDI track. Specifically, handcrafted (HC) features of the notes’ attributes are adopted as the metrics to select a track containing the main melody notes [12, 15, 16, 19–21, 31]. Traditional metrics utilized to filter out non-main melody notes include the entropy and average of pitch, the standard deviation of duration [32]. In addition to setting the range or threshold value of the metrics directly [32], the HC features could be used by some machine learning classifiers, including random forest [19, 33] and Naive Bayes classifier [22, 34]. Unlike the methods using HC features, Li et al. [35] try to calculate the similarity between tracks using different versions of a piece of music. Furthermore, similarity neighborhood [36] could be utilized to extract bars that contain main melody notes. However, the aforementioned track level-based methods require that the notes are located on separated MIDI tracks. If all the notes appear on the same track or the main melody notes are scattered on different tracks, these methods will fail [37].

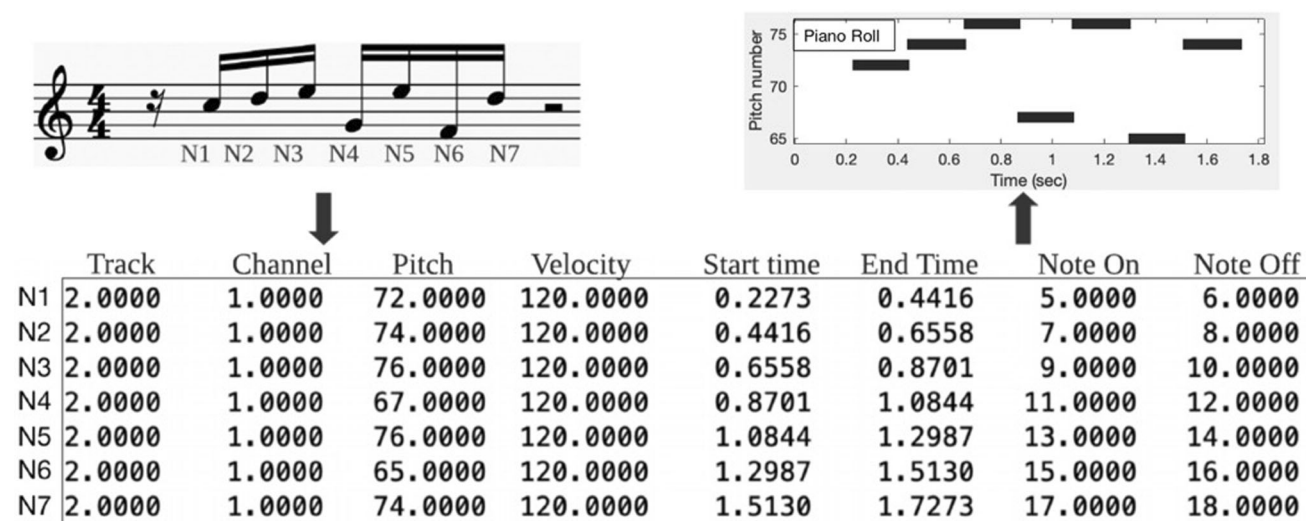


Fig. 2 Notes’ messages in MIDI file and the visualization of MIDI file (piano roll)

Note level methods Instead of using the features of the MIDI file, Uitdenbogerd et al. put forward a method named Skyline [15], which assumes that all the notes with the highest pitch at a time are the main melody notes. Subsequently, other researchers [13, 34] usually use the skyline algorithm to preprocess MIDI files to filter out part of the accompaniment, and they make improvements on the main melody extraction based on the preprocessed MIDI file. Later on, some non-machine learning methods identify the main melody by extracting repeated segments [38] or directly specifying a threshold of notes interval to filter out accompaniment notes [26]. Such methods are both inefficient and error prone (e.g., extracting the accompaniment part by mistake). Hence, researchers put forward intelligent methods using the machine learning technique in recent years, for example, the approach using LSTM [39], CNN model [13, 40] or Markov Chain [31]. In recent years, MIDiBert [24] and LSToM [25], which are large parameter model and small parameter model, respectively, that have obtained state-of-art results for main melody notes classification.

However, all the aforementioned methods assume that there is only one main melody. In many cases, it is difficult to determine only one main melody from the music, for example, see Fig. 3. Therefore, the aforementioned methods have significant limitations. Taking a different approach, Dai et al. [10] propose a framework based on Transformer-LSTM to generate new melody using basic melody and other music information. The basic melody could be understood as the main melody. Because Dai et al. [10] regard the basic melody extraction as a prediction task, this method could predict multiple melodies. However, the accuracy of the predicted main melody is around 39%, which means that the identified melody is not related to the corresponding music, and hence, it is not the main melody.

2.2.2 Audio-based methods

Some audio files are performance recordings. Others are converted from MIDI files which are usually clean without background noise. Audio-based main melody extraction only focuses on the former. Many conventional approaches estimate the pitch of main melody notes based on pitch salience and spectrogram [41–45]. These conventional methods are often complex and have many steps. For example, to extract the main melody, [42] first uses short-time Fourier transform (STFT) technology to obtain spectral peaks and corrects the frequency and amplitude of the music signal. Next, it obtains a group of melody candidates via the computed salience function. Finally, it will select a melody using some music features (e.g., pitch mean and deviation). This complicated method could get relatively good results, with 70% accuracy on average. In recent years, some methods have started to use the machine learning technique, like CNN [46, 47] or SVM [48]. Such machine learning-based methods is more accurate than traditional methods. However, the accuracy results are erratic. The best performing method above [47] ranges from an average of 75% to a maximum of 93%.

Table 1 tabulates the conventional methods designed for main melody extraction for both MIDI-based and audio-based approaches. Although most existing methods are aware of the problem of ambiguity in the main melody definition, they choose to simplify the problem, i.e., working under the existed dataset providing one main melody label. In addition, no matter the audio-based or midi-based approaches, machine learning techniques can usually significantly improve the accuracy of main melody extraction. Motivated by the aforementioned analysis, in this paper, we put forward a new definition of *main melody* and design a deep learning method for main melody extraction.

(a) Choral music
(b) Pop music

Fig. 3 Two examples of the music contain multiple main melodies: **a** is a segment of choral music (the 10th file in the Bach10 dataset [23]). **b** is a segment of pop music (the 874th file in POP909 dataset [4])

Table 1 Summary of existing main melody extraction works, including MIDI-based and audio-based

First Author	TL	MeL	NL	Non-ML	ML
<i>MIDI-based methods</i>					
Uitdenbogerd [15], Wei [16]	✓	–	✓	✓	–
Ozcan [12], Friberg [32], Li [35]	✓	–	–	✓	–
Martin [21], Chen [22], Rizo [19], Velusamy [20]	✓	–	–	–	✓
Jiang [34], Adiloglu [36]	–	✓	–	✓	–
Conklin [38], Wen [26]	–	–	✓	✓	–
Dai [10], Li [40], Zhao [31], Chou [24], Simonetta [13], Kosta [25]	–	–	✓	–	✓
<i>Audio-based methods</i>					
Salamon [42], Choi [48], Wu [47], Lee [46]	–	–	–	–	✓
Zhang [41], Paiva [45], Frieler [43]	–	–	–	✓	–

Note The task of extracting the main melody from MIDI will be categorized into three levels, track level (TL), measure level (MeL) and note level (NL). In addition, both MIDI-based and audio-based methods will be categorized by using machine learning (ML) and non-machine learning (Non-ML)

3 Methodology

Firstly, we put forward a new definition of the *main melody* for a music. Subsequently, we explore compound word as the data representation for further analysis, and finally, we detail the proposed Multi-MMLG (Multiple Main Melodies Generator) framework.

3.1 Main melody

When referring to the main melody of a music, the definition agreed upon by most articles is that melody is a unique sequence of notes used in identifying music [15, 34, 49]. In addition, main melody has also been defined as the part hummed and remembered by people, and it is the most attractive part of the music [49]. However, the above definitions of main melody are vague because different people have different perceptions and judgments [50, 51]. Hence, current definitions of the main melody are limited, and no single definition is adequate [49, 52]. For this phenomenon, many studies will specifically clarify the definition of the main melody to apply to their research field. Bittner et al. [52] use three main melody definitions in its research, arguing that multiple main melodies' definition is the most complex but the most general. Sequentially, we will discuss the complex case.

Main melodies in music are diverse, complex, and variable. Specifically, music usually contains multiple main melodies which are equally important [27]. For example, Fig. 3a shows a segment of choral music with two main melodies, namely Melody 1 and Melody 2. These main melodies have similar pitch contour in different keys, and Melody 1 contains more notes. It is a difficult task to decide which melody line is not the main melody

because both two melodies could be used to identify the music. Another example (pop music) is shown in Fig. 3b. Here, the two main melodies in the music are similar but have some different notes and number of notes. Therefore, the assumption of uniqueness in main melody adopted by existing research cannot be established, and it oversimplifies the problem of main melody extraction [49, 51].

Therefore, in this work, we put forward a new definition for main melody. For a piece of music, its main melodies should be relevant to the music so that they can be used to identify it. Furthermore, one music's main melodies are not entirely identical, such as in Fig. 3. In the event that the main melodies are identical, they are treated as single main melody, but not multiple main melodies. However, if the main melodies are substantially different, they may belong to different music. Hence, *main melody* could be defined as a set of similar but non-identical melodies that can be utilized to identify the music by entities (humans and computers). Subsequently, we predict multiple main melodies based on this new definition for main melody.

3.2 Data representation

We use Compound Word (CP) [53] to represent notes as the input of MIDIXLNet used in the framework's Stage 1. Each compound word contains five tokens, including Sub-Beat, Pitch, Duration, Chord, and Bar. Specifically, Sub-Beat is equivalent to a note's position in a bar, and the Pitch events are the absolute pitch value. For a piece of piano music, the corresponding MIDI number of absolute pitch values range from 21 to 108. To represent the chord events, we use a chord recognition method proposed by Huang and Yang [54], which basically uses a sliding window and proposes a rule of calculating chord

likelihood. After extracting the chords in a bar, we assign the recognized chord’s name to the notes that make up the chord. In addition, the Bar event is described as a binary number, namely 0 for a new bar and 1 for a continuing bar. By adopting the representation, each compound word has exactly five tokens. An example of the MIDIXLNet input is shown in Stage 1 of Fig. 4.

On the other hand, the original MuseBert [28] uses three notes’ events, including Onset, Pitch, and Duration, to describe melody. Onset event is equivalent to Sub-Beat, and both are the note’s position in a bar. In the modified MuseBERT model used in the framework’s Stage 1, we additionally use the notes’ track label (0/1), which specifies that the notes belong to the main/non-main melody track. Interested reader may refer to [28] for detailed information on MuseBert data representation.

3.3 Multi-MMLG framework

Our proposed Multi-MMLG (Multiple Main Melodies Generator) framework consists of 2 stages, namely a classification stage (Multi-MMLG_{s1}) and a conditional prediction stage (Multi-MMLG_{s2}). Figure 4 shows the structure of the Multi-MMLG framework. Both stages are detailed in the following subsections.

3.3.1 Stage 1: Multi-MMLG_{s1} & MIDIXLNet

The first stage in the proposed Multi-MMLG framework is the note level classification task. Using the classified results with high accuracy as the Stage 2’s condition guarantees that Stage 2 can learn suitable features from the condition, thus generating potential main melodies. Hence, we

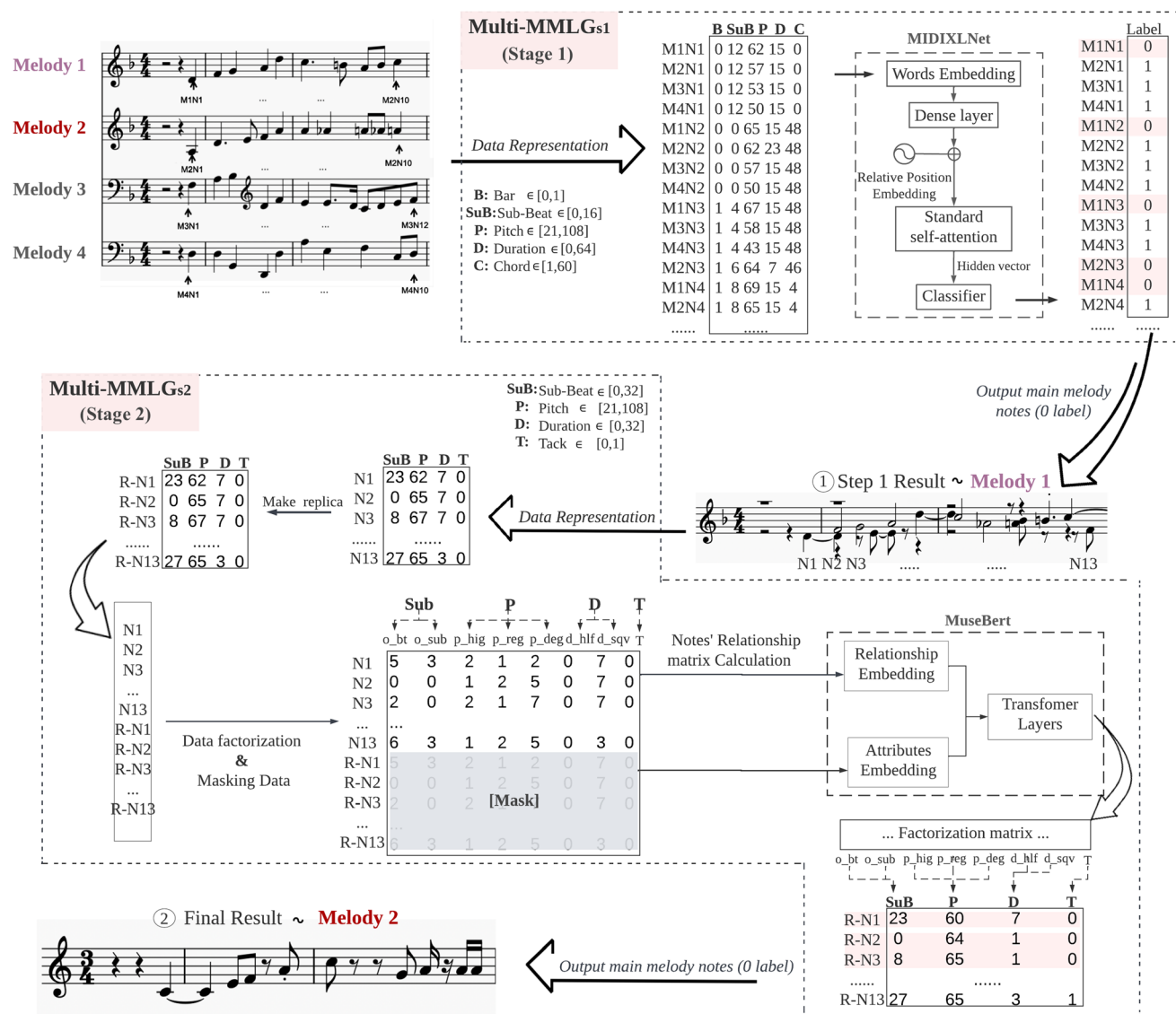


Fig. 4 Multi-MMLG Architecture: A pipeline architecture with two stages that could predict potential main melodies

implement a MIDIXLNet model to improve the accuracy of classifying main/non-main melody notes.

Figure 5 shows the pre-training structure of the MIDIXLNet model. Here, we would not use the notes' binary labels (i.e., main melody note and accompaniment) provided in the dataset during pre-training, which is an unsupervised learning task. After representing the notes, the prepared compound words will be embedded in the word embedding layers. Considering that each note is represented by five tokens, the embedding layer is five dimensions. All the embedded notes will be merged into one dimension matrix via a linear layer. At the same time, the words' position will be embedded following the standard relative positional encoding strategy used in the Transformer-XL model [55, 56].

Next, these embeddings will be fed to self-attention layers. The XLNet-based model uses a particular self-attention strategy, namely the Two-Stream self-attention strategy, including the content stream attention and query stream attention. This strategy could avoid the significant weakness of prevailing Masked-Language Modeling, namely directly corrupting input data so that some context features of music will be lost. Due to the contextual information of music being important [30, 57, 58], the advantage of avoiding loss of information is essential and is the reason behind why we implement an XLNet-based model. In addition, the permutation of likelihood factorization's order will support the XLNet-based model to capture bidirectional context. Hence, we adopt the permutation mechanism and the two-stream self-attention strategy.

Specifically, the permutation is used to change the factorization order of likelihood rather than the sequence order of the notes. Since the permutation generates many

possible results (i.e., $n!$), Eq. (1) will be used to sample a permutation set z .

$$\max_{\theta} : E_z \sim Z_n \left[\sum_{p=1}^n \log p_{\theta}(N_{zp} | M_{z < p}) \right]. \tag{1}$$

In Eq. (1), Z_n represents all permutation results of the likelihood factorization order for the melody sequence M with length n . In addition, N_{zp} represents the word of the note at position p in the sampling permutation z set, and $M_{z < p}$ refers to the notes before position p in z set. Subsequently, the likelihood p_{θ} of the sampling order will be calculated.

On the other hand, the content stream attention (a.k.a. standard self-attention) could provide information of the current token and the preceding tokens. It is expressed as follows:

$$h_{zp}^m = \text{Attention} \left(Q = h_{zp}^{m-1}, KV = h_{z \leq p}^{m-1} \right), \tag{2}$$

where Q , K , and V are vectors for Query, Key, Value, respectively, and zp is the target token's position following z set's likelihood factorization order. In layer $m - 1$, Q knows the target's content and position (h_{zp}^{m-1}). KV will know the content and position information of the target token and the tokens before position p ($h_{z \leq p}$) in z . Note that the query stream attention uses the binary number to encode whether the target's position could be queried or not. Meanwhile, it also encodes that the position and content of tokens before the target could be queried or not [55]. This mechanism is used to avoid accessing the target's content during querying. The expression is written as follows:

$$g_{zp}^m = \text{Attention}(Q = g_{zp}^{m-1}, KV = h_{z < p}^{m-1}) \tag{3}$$

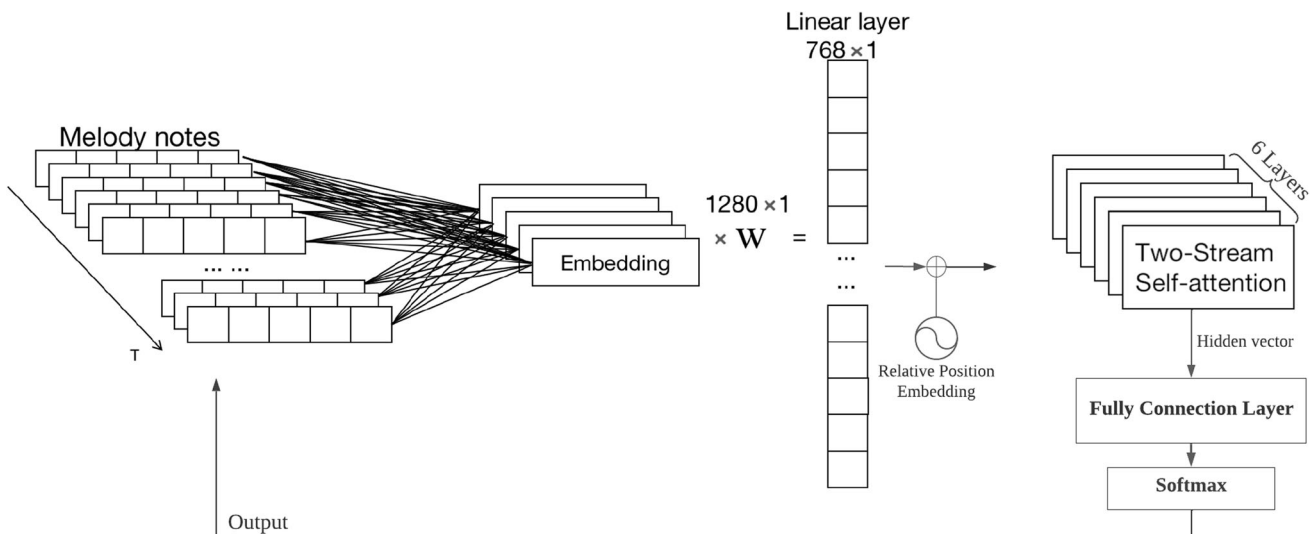


Fig. 5 Pre-training procedure of MIDIXLNet

Equation (3) is similar to Eq. (2), but the difference is that query stream attention will access the token before the target ($h_{z < p}$). Furthermore, Q will only know the target token's position (g_{zp}) rather than both position and content (h_{zp}).

Figure 6a shows a melody segment, and Fig. 6b shows the melody's corresponding matrices of Two-Stream self-attention before permutating the likelihood factorization order. The melody sequence could be written as $M = (N_1, \dots, N_n)$, where N_1 refers to the first note and the length $n = 5$. In order to predict N_2 following the original factorization order, only N_1 could be accessed before the permutation, i.e., is $P(N_2 | N_1)$, which is uni-directional learning.

Figure 7 shows the Two-Stream self-attention matrices after sampling permutation order. Under this case, when predicting N_2 , $h_{z \leq 4}$ includes the position and content of N_5, N_4 , and N_1 rather than only N_1 .

Furthermore, our architecture contains 6 self-attention layers. After mapping the hidden vector from the attention layers to fully connection layers, the prediction results are calculated using a standard softmax function on its normalized value.

When using the MIDIXLNet model for fine-tuning, the standard self-attention will be used rather than Two-Stream self-attention. The fine-tuning framework is shown in Stage 1 in Fig. 4. In addition, the binary classifier used to classify main/non-main melody notes is similar to the classifier in the MIDIBERT method using ReLU as the activation function [24].

3.3.2 Stage 2: Multi-MMLG_{s2}

The second stage of the Multi-MMLG framework will predict multiple main melodies. Recall that the potential main melodies of a piece of music are similar but non-identical (see Sect. 3.1). The MuseBERT model proposed by Wang and Xia [28] aims to predict new melodies which do not conform to our main melody definition. Hence, we modified the model to make it more suitable for main melody extraction.

The structure of the modified MuseBERT model is shown (enclosed in a box with dotted lines) in Fig. 4. First, we make a replica of the classification results from Stage 1. Next, all notes in the replica will be masked and will be finally predicted. Under this case, there are two conditions for predicting, Stage 1's output and the notes relationship matrices. In other words, the masked notes will be predicted based on these two conditions. This prediction strategy is different from the original MuseBert [28] that directly masks all music notes and only uses one condition (i.e., the relationship among notes) to predict masked notes.

After preparing the data, all input is mapped to twelve linear embedding layers with eight attention heads. Finally, the last hidden vector will be normalized to generate the results via the softmax layer.

The pre-training stage of the modified MuseBERT model is similar to the prediction stage. The only difference is that we do not make a replica of the input, and we mask part of the notes for training. Hence, in the pre-training stage, our conditions are the relationship among the notes and the unmasked notes.

Before producing the predicted results, there is still a problem with the original MuseBert [28] model. It is a BERT-based model, and it cannot predict the next notes freely, meaning that it can only predict the masked notes. However, the predicted masked notes may not belong to the main melody. Hence, as mentioned in Sect. 3.2, we add the track label of every note. Under this setting, the modified MuseBERT model could extract the note whose track label is predicted as 0 (0: main melody notes), thus filtering out the non-main melody notes.

In summary, Multi-MMLG_{s2} adopts the modified MuseBERT model, which uses a different prediction strategy and improve the flexibility of predicting the main melody by adding the track label of the notes.

4 Experiments

4.1 Experiment settings

For evaluation purposes, we use the Mixed POP909 dataset, which includes POP909_m (MELODY track's notes are the main melody) and POP909_{mb} (MELODY track and BRIDGE track's notes are the main melody). An example that illustrates the Mixed POP909 dataset is shown in Fig. 8.

In the Stage 1 (i.e., Multi-MMLG_{s1}) using the MIDIXLNet model to classify main melody notes, we split the Mixed POP909 dataset into 8:1:1 for training:validation:testing purposes. Our model is trained on an RTX3060 GPU with 6GB. Since this GPU has less memory, we set the batch size to 1, and the maximum length is 512. We train the model for 45 epochs, and each epoch takes about 10 min. During the fine-tuning process where only the standard self-attention is used, each epoch takes about 5 min, and we also train for 45 epochs.

For the second stage (i.e., Multi-MMLG_{s2}), using the modified MuseBERT model to conduct conditional prediction, we also split Mixed POP909 into 8:1:1 for training:validation:testing purposes. As discussed in Sect. 3.3.2, the training stage does not need to make a replica of Stage 1's output as the condition, so the data pre-processing strategy of the final prediction and the training stages

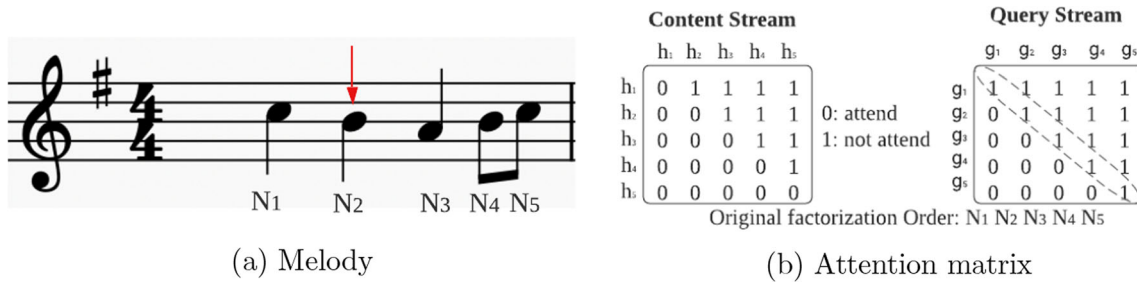


Fig. 6 A melody example (a) and its corresponding attention matrices (b)

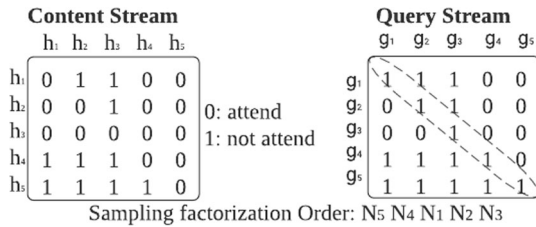


Fig. 7 Two-Stream self-attention matrices after sampling permutation order

have a few differences, as shown in Fig. 9. During training, 15% of the data will be corrupted following a specific strategy in Fig. 9a. In the final prediction stage, we use a random masking strategy (0%–45%) on Stage 1’s output. As shown in Fig. 9b, 0% masking rate of the Stage 1’s (Multi-MMLG_{s1}) output means that the modified

MuseBERT model needs to learn the whole output to predict the main melodies. Generally, using the conditions with more notes (e.g., 0% masking rate) will generate melodies that are highly similar to the corresponding music. Conversely, as the masking rate increases, there are fewer and fewer notes in condition. The similarity between the predicted melody from the model that has fewer conditions and the corresponding music will decrease. Hence, the 0%–45% masking rate aims to ensure that the similarity between the predicted main melodies and the corresponding music will neither be too high nor too low, thus conforming to our new main melody definition. Essentially, the masking rate is varied to manage the condition, thus generating melody. This managing condition strategy is similar to the method used by Hadjeres and Nielsen [59],

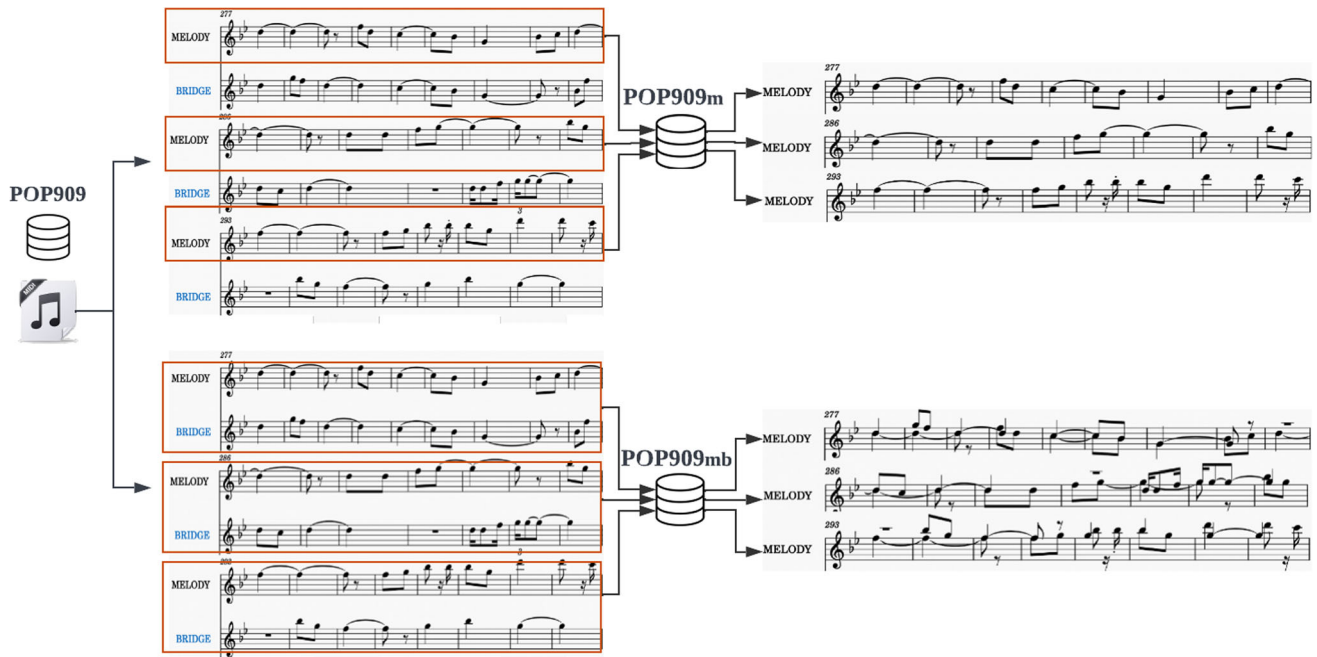
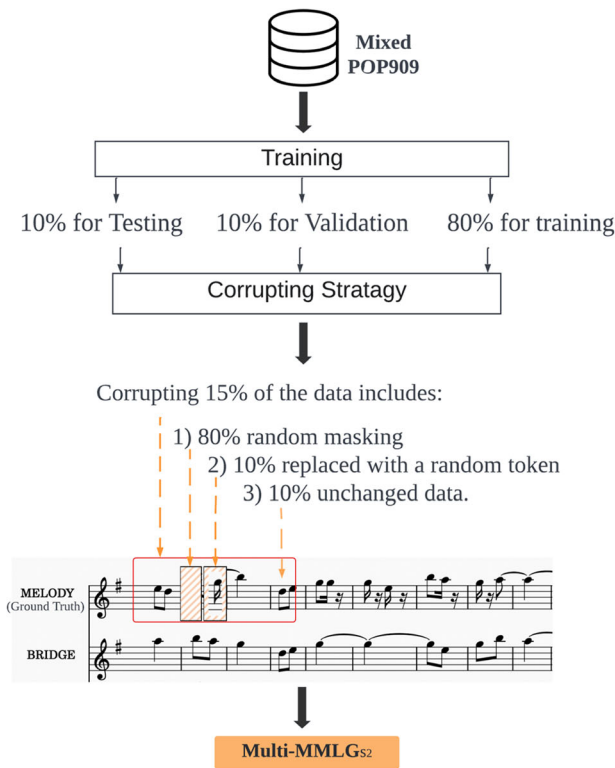
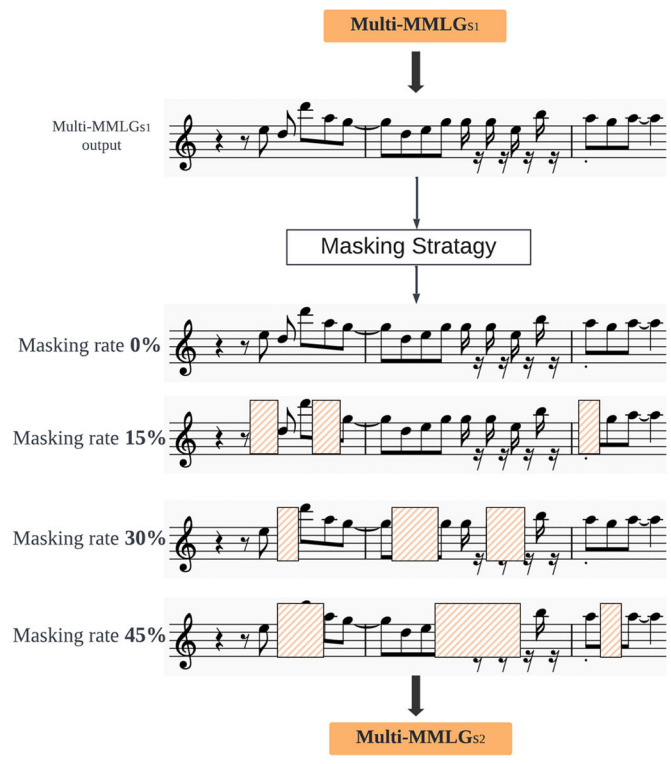


Fig. 8 Main melody of 808th file in Mixed POP909 dataset (POP909_m: regarding the MELODY track as the main melody. POP909_{mb}: regarding the MELODY and BRIDGE tracks’ notes as the main melody)



(a) Corrupting strategy



(b) Random Masking

Fig. 9 Examples illustrate the different data pre-processing strategies during the training and prediction. Corrupting strategy used in the training stage is shown in (a), and the random masking strategy used

on the final prediction’s conditions of Multi-MMLG_{s2} is shown in (b). The part covered by the twill-filled rectangle represents the randomly selected masking notes

which directly specifies notes as the condition, thus creating new melodies.

4.2 Results and analysis

This section conducts a comprehensive evaluation of the proposed Multi-MMLG framework. We first evaluate the effectiveness of the Multi-MMLG framework’s Stage 1 using the proposed MIDIXLNet model and compare its performance with the other four models. Subsequently, we perform an ablation experiment on MIDIXLNet to explore the influence of the individual components in MIDIXLNet. In addition, we also verify the necessity of designing a model specifically for the midi-based main melody classification task. Next, we conduct an ablation experiment on the Multi-MMLG framework. These experiments can justify the selected models within the framework, verifying the influence on the final predicted results when using different models in Stage 1 of the framework, and comparing the predicted model in Stage 2 with four models. Finally, we evaluate the potential main melodies predicted by the proposed Muti-MMLG framework under different masking rates.

4.2.1 Stage 1: Multi-MMLG_{s1} & MIDIXLNet

Recall that the MIDIXLNet is the model; we proposed and used in the Stage 1 in Multi-MMLG framework (i.e., Multi-MMLG_{s1}). Firstly, we adopt *Accuracy*¹ and *Parameter Number* (PN) to evaluate the effectiveness and efficiency of MIDIXLNet based on the Mixed POP909 dataset (POP909_{mix}). Table 2 shows the results of comparing the MIDIXLNet model with conventional models. For all compared models, the RNN-based model with one classifying layer (RNN_{cl}) is set as the baseline model. In addition, we also compare the proposed MIDIXLNet model to MIDIBert [24] and LSToM [25], which are large parameter model and small parameter model, respectively, that have obtained state-of-art results in recent years. Note that the models in each group in Table 2 have the same data description approach.

It can be seen from Table 2 that for the complex POP909_{mix} dataset, MIDIBert [24] and LSToM [25] achieve no more than 90% accuracy, suggesting some

¹ Accuracy is “the total number of predicted correct music notes” divided by “the total number of predicted incorrect music notes.”

Table 2 Comparison of using different main melody classification models

	RNN _{cl}	MIDIBert [24]	MIDIXLNet	MIDIBert ₅	LStoM [25]
Accuracy	86.73%	89.62%	97.37%	96.84%	88.20%
PN ($\times 10^6$)	6.1	111.2	71.8	111.4	2.5

The best results are highlighted in boldface

RNN_{cl} and MIDIBert [24] adopt MIDIBert [24]'s four dimensions description approach, while MIDIXLNet and MIDIBert₅ adopt the five dimensions description approach described in Sect. 3.2. LStoM [25] uses its six dimensions description approach. Additionally, RNN_{cl} is RNN with a single classifying layer

room for improvement. Furthermore, the result of RNN_{cl}, which has the same data description method as MIDIBert [24], is lower. However, based on our proposed 5-dimensional data input (including Bar, Sub-Beat, Position, Duration, and Chord), after restructuring MIDIBert [24] (MIDIBert₅), the result shows a significant improvement, i.e., 96.84% accuracy. Under the same data description method as MIDIBert₅, the proposed MIDIXLNet can achieve a higher accuracy of 97.37%, while its parameter size is approximately 64% of that of MIDIBert₅.

Next, we perform an ablation experiment to explore the influence of the individual components within MIDIXLNet, and the results are recorded in Table 3. Specifically, when the masking strategy used in the XLNet model is similar to that of the Bert model (rather than using the unique two-stream self-attention mechanism of the XLNet model), the performance of the main melody classification drops to 96.81%. In other words, under the premise of fewer parameters than the MIDIBert₅, the results obtained without the two-stream self-attention mechanism are very close to those of MIDIBert₅ in Table 2. Also, the reduction in the number of hidden layers appears to have the most apparently negative impact on performance. The other contributing factor to the performance deterioration is the change of the attention type from bi-direction to uni-direction.

Comparison with non-main melody classification approach

In this section, we conduct two experiments to justify the necessity of designing a new model for the main melody classification task based on MIDI files.

First, recall from earlier that music is usually recorded as the audio file (e.g., WAV) or the symbolic file (e.g., MIDI). To verify the importance of designing the main melody classification method for MIDI files, we evaluated the result of the main melody extraction method for audio. To be more specific, we use the midi2audio package to convert the MIDI files in the Mixed POP909 dataset to audio files. The algorithm for extracting the main melody from audio and output to MIDI files is based on the Melodia algorithm² [42]. To evaluate and compare the

results, we use edit_distance [36, 60, 60–62] and Dynamic Time Warping (DTW) [1, 60]. The edit_distance can calculate the similarity between sequences, and it is robust against different lengths [60]. Hence, the edit_distance metric is used to calculate pitch similarity (PS) and chord similarity (CS) between the extracted main melody and ground truth. On the other hand, DTW, also referred to as melody distance (MD) by Tsai et al. [1] and Ju et al. [60], is an indicator based on dynamic programming. It takes into account the pitch value and duration value together. Table 4 records the results of the aforementioned metrics for both Melodia [42] and our proposed MIDI-based method (i.e., MIDIXLNet). Results suggest that the melodies extracted from the audio file by Melodia [42] are not the main melody due to the low similarity and large MD values. The main reason for the non-ideal results is that the audio files converted from MIDI files lose an important perceptual feature, namely loudness. In other words, the feature that *the main melody is usually louder than other melodies in music* is the key feature used to extract the main melody from the audio file, but that is not the main feature in the MIDI file. As a result, the main melody from a MIDI file cannot be extracted accurately using the audio-based method due to the differences between a MIDI file and an audio file. In short, when considering the results collected for MIDIXLNet, it is apparent that MIDIXLNet outperforms Melodia [42] for the task of classification of main melodies for all metrics considered in this work. Therefore, it is necessary to design the MIDI-based approach.

Secondly, Table 5 records the performances achieved by cross-domain transfer learning, which further justifies the need to design a model specifically for the main melody classification task. This experiment is inspired by Bukhsh et al. [63]'s promising results via cross-domain transfer learning strategy. Specifically, considering that, in recent years, Wu et al. [64] obtained better results in the field of Few-shot object detection, and we refer to their base detector model to conduct the cross-domain transfer learning, namely the Faster R-CNN model [65]. In addition, the backbone of Faster R-CNN is Resnet-101 [66], which is a popular architecture used with Faster R-CNN [64, 67]. In fact, Faster R-CNN [65] aims to detect objects

² An approach that extracts main melody from audio and outputting to MIDI file https://github.com/justinsalomon/audio_to_midi_melodia.

Table 3 Ablation experiment for the proposed MIDIXLNet model using the POP909_{mix} dataset

Ablation	Accuracy (%)
<i>MIDIXLNet</i>	97.37
—Without Two-Stream Self-Attention mechanism	96.81
–Attention type (attn_type = ‘uni’)	95.44
–Attention head (n_head = 2)	96.53
–Hidden layers (n_layer = 2)	93.87

The best results are highlighted in boldface

Table 4 Comparison of the proposed MIDI-based main melody classification method (i.e., MIDIXLNet) with the audio-based method (i.e., Melodia)

Models	POP909 _m			POP909 _{mb}		
	PS	CS	MD	PS	CS	MD
Melodia [42]	26.9%	27.0%	3.21	30.1%	21.6%	3.79
MIDIXLNet	98.2%	94.2%	0.82	98.6%	95.8%	0.36

The best results are highlighted in boldface

The Pitch Similarity (PS), Chord Similarity (CS), and Melody Distance (MD) are recorded

Table 5 Performances attained by Transfer learning based on ResNet-101 using Faster R-CNN model

	POP909 _{mb}	POP909 _m	POP909 _{mix}
Faster R-CNN [65]	43.67%	45.07%	43.69%
MIDIXLNet	98.34%	96.45%	97.37%

The best results are highlighted in boldface

in an image. Hence, in our input matrix, we set the ground truth of the object region as 1 (width) × 5 (height), which means there are 512 regions in total in the input 512 × 5 matrix. Based on the results recorded in Table 5, cross-domain transfer learning, in its current form, has not achieved promising results. Specifically, the accuracy is low (around 44%), which is significantly lower than the accuracy obtained by our method designed for the extraction of main melodies.

Overall, neither cross-domain transfer learning nor audio-based melody extraction can achieve good results in the field of midi-based main melody classification. Therefore, it is necessary to design a model specifically for the main melody classification task based on the midi file.

4.2.2 Stage 2: Multi-MMLG_{s2}

As per our newly proposed definition for main melody, it should be relevant to the corresponding music and similar

at a moderate level. Considering that the Mixed POP909 dataset contains the main melody ground truth, the melodies similar to the ground truth at a moderate level are the potential main melodies. Therefore, we use three metrics described in section 4.2.1, namely PS, CS and MD, to evaluate the similarity between the final predicted results and the main melody’s ground truth of the music.

Table 6 shows the evaluation results of ablation experiments of the Multi-MMLG framework. In this experiment, we do not evaluate Stage 1 or Stage 2 of our framework separately because these two stages cannot work independently to predict potential main melodies. In other words, the classification stage (i.e., Stage 1 in Multi-MMLG framework) can only classify one main melody rather than multiple main melodies, and the prediction stage (i.e., Stage 2 in Multi-MMLG framework) can only predict new melodies rather than main melodies. Hence, none of the stages can predict multiple main melodies independently as we have anticipated. Hence, we use different models to replace the two stages of the Multi-MMLG framework, thus justifying the selected models within the framework, including verifying the influence on the final predicted results when using different models in Stage 1 of the framework and comparing the predicted model in Stage 2 with four models.

For one of the control groups in Table 6, we use MuseBert [28], XLNet, RNN_{pre} and MusicTransformer [68] to replace the modified MuseBERT model used in Multi-MMLG_{s2}, i.e., MM_{s1} + MuseBert/XLNet/RNN_{pre}/MusicTransformer. These experiments compare the predicted results from the modified MuseBert with four other music-predicted methods, thus demonstrating the necessity of using the two predicting conditions (i.e., MIDIXLNet’s classified results from Multi-MMLG_{s1} and music notes’ relationship). Specifically, MM_{s1} + XLNet/RNN_{pre}/MusicTransformer [68] control groups use one condition, i.e., classified results from Multi-MMLG_{s1}. These three methods directly predict the next notes. Table 6 suggests that their predicted results show considerably low similarity (i.e., Both PS and CS are below 30%, and MD is larger than our proposed Multi-MMLG framework using MIDIXLNet and the modified MuseBERT models) with reference to the corresponding music. Hence, these three methods cannot predict the potential main melodies. Similarly, although MM_{s1} + MuseBert [28] uses another condition, i.e., notes’ relationship, and its predicted results are still not ideal. Overall, the proposed Multi-MMLG framework, which uses MIDIXLNet and a modified MuseBERT model in sequence and combines the above two conditions, could significantly improve all three-similarity metrics, i.e., PS, CS and MD.

Table 6 Ablation experiments' results of the Multi-MMLG framework

Models	POP909 _m			POP909 _{mb}		
	PS	CS	MD	PS	CS	MD
RNN _{cl} + MM _{s2}	51.1%	30.2%	3.37	68.0%	34.8%	4.79
MIDIBert [24] + MM _{s2}	58.9%	41.9%	3.25	71.5%	38.5%	4.41
MM _{s1} + RNN _{pre}	13.7%	14.3%	7.53	14.1%	14.7%	6.91
MM _{s1} + XLNet	18.0%	24.5%	4.72	19.7%	24.8%	4.40
MM _{s1} + MusicTransformer [68]	20.0%	26.3%	2.87	22.1%	30.2%	3.18
MM _{s1} + MuseBert [28]	10.9%	18.9%	9.6	14.5%	25.3%	4.59
Multi-MMLG	81.8%	62.4%	1.77	83.8%	63.3%	2.77

The best results are highlighted in boldface

Note This table, respectively, evaluates the effectiveness of using different classification methods in Stage 1 of Multi-MMLG framework and that of using different prediction methods in Stage 2

MM_{s1}—the first stage of Multi-MMLG framework, namely MIDIXLNet model.

MM_{s2}—the second stage of Multi-MMLG framework, namely the modified MuseBERT model.

RNN_{pre}—RNN model for prediction melody

For another control group, we use RNN_{cl} and MIDIBert [24] to replace MIDIXLNet used in the Multi-MMLG_{s1}, i.e., RNN_{cl}/MIDIBert [24] + MM_{s2}. In comparison with Multi-MMLG framework, this control group can justify using the MIDIXLNet model in Multi-MMLG_{s1} and can prove the importance of classified main melody's high accuracy. This control group also uses two conditions, i.e., classified melody and music notes' relationship. However, the predicted results are still worse than that of Multi-MMLG. From these results, we confirm that, in addition to using the above two conditions, the accuracy of one condition, namely classified melody, is the second factor that affects the final predicted results. Therefore, Multi-MMLG simultaneously responds to these two factors, thus achieving improved results.

In addition, as introduced in Sect. 4.1, we use the random masking rate to control the prediction's condition, thus controlling the predicted main melodies. Therefore, we also evaluate the results of the Multi-MMLG framework under different masking rates.

To be more specific, the potential main melody should be similar to the ground truth in a suitable range, neither too high nor too low. To control the final results, we perform a masking strategy (0%–45%) on Stage 1's output, as introduced in Sect. 4.1. The results are shown in Table 7. At a 0% masking rate, the results are closer to the ground truth, while for a higher masking rate (e.g., 45%), the results are less similar to the music. Hence, it is easy to infer that, as the masking rate increases, the model will predict main melodies with fewer notes in condition, and the predicted melodies will become more irrelevant to the corresponding music. However, Table 7 shows that the PS of the prediction results under 30% masking rate using the POP909_{mb} dataset is significantly better than the 15%

masking rate. After a comprehensive observation of the evaluation results, 15%–30% is a reasonable masking range for predicting the potential main melodies. At the same time, the results from using different masking rates can also prove the robustness of Multi-MMLG. In other words, even if the masking rate reaches 30% and the notes in the condition become less, the similarity between the main melodies predicted by the model, and the corresponding music is still better than other methods in Table 7.

In conclusion, the structure and the models selected in Multi-MMLG framework are reasonable, and this framework could output multiple main melodies.

4.2.3 Qualitative analysis

We perform a qualitative analysis for a more intuitive view of the predicted melodies. Figure 11 presents the results of embedding other models into our framework. Consistent with the quantitative analysis results, the predicted results are different from the Melody track or Bridge track. Hence, using the methods shown in Fig. 11 could not work. Figure 10 shows some results achieved by the Multi-MMLG framework when ranging the masking rate between 0% and 45%. In the Figs. 11 and 10, the musical scrolls containing the ground truth main melody is derived from the POP909_m dataset. This dataset only regards the MELODY track as the main melody. If the predicted results are similar and non-identical to the MELODY track, these results are the potential main melodies. In addition, the BRIDGE melody is also the main melody following our new main melody definition. Since POP909_m ignores this potential main melody, if the predicted results are similar to the BRIDGE

Table 7 Similarity between the generated and ground truth melodies under different masking rates (MR)

	MR (%)	POP909 _m			POP909 _{mb}		
		PS (%)	CS (%)	MD	PS (%)	CS (%)	MD
Multi-MMLG	0	81.1	62.4	1.77	83.8	63.3	2.77
	15	71.1	42.8	2.63	55.6	43.9	2.67
	30	66.6	44.5	3.02	72.0	40.2	5.08
	45	62.9	39.3	3.13	53.5	39.7	2.98



Fig. 10 Results of using the Multi-MMLG framework to predict the potential main melodies of the 874th file (in POP909_m dataset). Under 15%-30% masking rate of the condition, the predicted results are potential main melodies. Especially for the first result under 15% masking rate, it is similar to the BRIDGE melody that is one typical potential main melody ignored by POP909_m



Fig. 11 Results of the control group using other models to replace the two stages of the Multi-MMLG framework to predict the potential main melodies of the 874th file (in the POP909_m dataset), including MIDIBert [24], XLNet and MuseBert_o

melody, it could better justify that our framework successfully extracts the potential main melody.

According to the above analysis, in Fig. 10, the first result in 15% masking rate is more similar to the BRIDGE melody, which is a typical example that proves our

framework’s ability to predict the main melody. Consistent with the results of quantitative analysis, the predicted results for 0% masking rate contain less difference from the ground truth, and 45% result is irrelevant to the corresponding music. Therefore, when operating in the range between 15% and 30% of masking rate, the Multi-MMLG framework could predict the potential main melodies of a piece of music.

5 Conclusion

This paper first puts forward a new definition for main melody. It acknowledges the fact that the main melodies are not unique but instead they are a set of similar and non-identical melodies. This definition is complex but more suitable for the applications such as music information retrieval. We proposed a framework that addresses the problem of automatically predicting multiple main melodies, and it caters for the complexity and diversity of the main melody. The two stages pipeline framework involves the main melody classification stage and a conditional prediction stage. Specifically, MIDIXLNet used in Stage 1 of the Multi-MMLG framework is proposed to provide main melody classification results with high accuracy efficiently. High accuracy classified results could preserve more context features, which becomes the condition of predicting at Stage 2. To predict potential main melodies that are similar in a reasonable range, we conduct a masking strategy on Stage 2’s condition. Experiment results suggest that Multi-MMLG could efficiently obtain high-accuracy classification results of the main melody and automatically extract multiple potential main melodies when the masking rate fall in the range between 15 and 30%.

Moreover, the experimental results show that the chord feature significantly impacts the melody classification results, increasing the accuracy by around 6%. In addition, we observe that when the masking rate of the predicting condition varied, PS, CS, and MD do not completely decrease. In other words, as the conditions for predicting the main melody become more relaxed, the similarity between the predicted main melody and the corresponding music may be likely to be improved. This characteristic

benefits the application of the framework in the field of main melody completion.

Currently, this framework controls the predicted results by randomly masking the prediction conditions. However, random masking causes the prediction results not to adapt precisely to the application environment. In other words, the predicted results may randomly and irregularly generate the main melody with different music keys, similar strong beats, or similar weak beats. In this case, when people expect the predicted main melodies to have a particular structure, they need to predict them multiple times. Therefore, in the future, we will consider adjusting the framework's Stage 2. We would use multiple relationship matrices, such as the strong and weak beat relationship matrix, the chord relationship matrix, as the input of the modified MuseBERT model replacing random masking. After specifying the required relationship matrix, our framework will be able to predict the main melody that meets human requirements more directly. This adjustment involves two parts, designing the representation of different relationship matrices and embedding methods dealing with different inputs.

Acknowledgements This work was supported by the Advanced Engineering Platform's Cluster Funding (account number AEP-2021-Cluster-04), Monash University Malaysia, Malaysia.

Funding Open Access funding enabled and organized by CAUL and its Member Institutions.

Data availability The datasets analyzed during the current study are available in the Google drive repository—<https://tinyurl.com/mw8k5xtx>

Declarations

Conflict of interest We declare that we have no financial and personal relationship with other people or organizations that can inappropriately influence our work, and there is no professional or other personal interest of any nature or kind in any product, service and/or company that could be construed as influencing the position presented in, or the review of, the manuscript entitled.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Tsai W-H, Yu H-M, Wang H-M, Horng J-T (2008) Using the similarity of main melodies to identify cover versions of popular songs for music document retrieval. *J Inf Sci Eng* 24(6):1669–1687
2. Simonetta F, Ntalampiras S, Avanzini F (2019) Multimodal music information processing and retrieval: survey and future challenges. In: International workshop on multilayer music representation and processing (MMRP). IEEE, pp 10–18
3. Ren Y, He J, Tan X, Qin T, Zhao Z, Liu T-Y (2020) Popmag: pop music accompaniment generation. In: Proceedings of the 28th ACM international conference on multimedia, pp 1198–1206
4. Wang Z, Chen K, Jiang J, Zhang Y, Xu M, Dai S, Gu X, Xia G (2020) Pop909: a pop-song dataset for music arrangement generation. arXiv preprint [arXiv:2008.07142](https://arxiv.org/abs/2008.07142)
5. He T, Liu W, Gong C, Yan J, Zhang N (2021) Music plagiarism detection via bipartite graph matching. arXiv preprint [arXiv:2107.09889](https://arxiv.org/abs/2107.09889)
6. Robine M, Hanna P, Ferraro P, Allali J (2007) Adaptation of string matching algorithms for identification of near-duplicate music documents. In: Workshop on plagiarism analysis, authorship identification, and near-duplicate detection (PAN07), pp 37–43
7. Cheng Y, Chen X, Yang D, Xu X (2017) Effective music feature ncp: enhancing cover song recognition with music transcription. In: Proceedings of the 40th international ACM SIGIR conference on research and development in information retrieval, pp 925–928
8. Tsai W-H, Yu H-M, Wang H-M, Horng J-T (2008) Using the similarity of main melodies to identify cover versions of popular songs for music document retrieval. *J Inf Sci Eng* 24(6):1669–1687
9. Teng Y, Zhao A, Goudeseune C (2017) Generating nontrivial melodies for music as a service. arXiv preprint [arXiv:1710.02280](https://arxiv.org/abs/1710.02280)
10. Dai S, Jin Z, Gomes C, Dannenberg RB (2021) Controllable deep melody generation via hierarchical music structure representation. arXiv preprint [arXiv:2109.00663](https://arxiv.org/abs/2109.00663)
11. Shih Y-J, Wu S-L, Zalkow F, Müller M, Yang Y-H (2021) Theme transformer: symbolic music generation with theme-conditioned transformer. arXiv preprint [arXiv:2111.04093](https://arxiv.org/abs/2111.04093)
12. Ozcan G, Isikhan C, Alpkocak A (2005) Melody extraction on midi music files. In: Seventh IEEE international symposium on multimedia (ISM'05). IEEE, p. 8
13. Simonetta F, Cancino-Chacón C, Ntalampiras S, Widmer G (2019) A convolutional approach to melody line identification in symbolic scores. arXiv preprint [arXiv:1906.10547](https://arxiv.org/abs/1906.10547)
14. Raposo F A, Martins de Matos D, Ribeiro R (2021) Assessing kinetic meaning of music and dance via deep cross-modal retrieval. *Neural Comput Appl* 33(21):14 481-14 493
15. Uitdenbogerd AL, Zobel J (1998) Manipulation of music for melody matching. In: Proceedings of the sixth ACM international conference on Multimedia, pp 235–240
16. Wei Z, Xiaoli L, Yang L (2014) Extraction and evaluation model for the basic characteristics of midi file music. In: The 26th Chinese control and decision conference, CCDC. IEEE pp. 2083–2087
17. Dannenberg RB (2006) The interpretation of midi velocity. In: ICMC
18. Briot J-P (2021) From artificial neural networks to deep learning for music generation: history, concepts and trends. *Neural Comput Appl* 33(1):39–65
19. Rizo D, De Leon PJP, Pertusa A, Pérez-Sancho C, Quereda JMI (2006) Melody track identification in music symbolic files. In: FLAIRS conference, pp 254–259

20. Velusamy S, Thoshkahna B, Ramakrishnan K (2007) A novel melody line identification algorithm for polyphonic midi music. In: International conference on multimedia modeling. Springer, pp 248–257
21. Martín R, Mollineda RA, García V (2009) Melodic track identification in midi files considering the imbalanced context. In: Iberian conference on pattern recognition and image analysis. Springer, pp 489–496
22. Chen L, Ma YJ, Zhang J, Wan GC, Tong MS (2018) A novel extraction method for melodic features from midi files based on probabilistic graphical models. In: Progress in electromagnetics research symposium (PIERS-Toyama). IEEE, pp 729–733
23. Duan Z, Pardo B, Zhang C (2010) Multiple fundamental frequency estimation by modeling spectral peaks and non-peak regions. IEEE Trans Audio Speech Lang Process 18(8):2121–2133
24. Chou Y-H, Chen I, Chang C-J, Ching J, Yang Y-H et al. (2021) Midibert-piano: large-scale pre-training for symbolic music understanding. arXiv preprint [arXiv:2107.05223](https://arxiv.org/abs/2107.05223)
25. Kosta K, Lu WT, Medeot G, Chanquion P (2022) A deep learning method for melody extraction from a polyphonic symbolic music representation. In: Ismir 2022 hybrid conference
26. Wen R, Chen K, Xu K, Zhang Y, Wu J (2019) Music main melody extraction by an interval pattern recognition algorithm. In: Chinese control conference (CCC). IEEE, pp 7728–7733
27. Fujioka T, Trainor LJ, Ross B, Kakigi R, Pantev C (2005) Automatic encoding of polyphonic melodies in musicians and nonmusicians. J Cognit Neurosci 17(10):1578–1592
28. Wang Z, Xia G. (2021) Musebert: pre-training music representation for music understanding and controllable generation. In: Proceedings of the 22nd international society for music information retrieval conference. Online: ISMIR, pp 722–729. [Online]. Available: <https://doi.org/10.5072/zenodo.940538>
29. Sharma A, Sharma K, Kumar A (2022) Real-time emotional health detection using fine-tuned transfer networks with multimodal fusion. Neural Comput Appl. <https://doi.org/10.1007/s00521-022-06913-2>
30. Oore S, Simon I, Dieleman S, Eck D, Simonyan K (2020) This time with feeling: learning expressive musical performance. Neural Comput Appl 32(4):955–967
31. Zhao H, Qin Z (2014) Tunerank model for main melody extraction from multi-part musical scores. In: 2014 sixth international conference on intelligent human-machine systems and cybernetics, vol. 2. IEEE, pp 176–180
32. Friberg A, Ahlback S (2009) Recognition of the main melody in a polyphonic symbolic score using perceptual knowledge. J New Music Res 38(2):155–169
33. Bittner R, Salamon J, Essid S, Bello J (2015) Melody extraction by contour classification. In: International conference on music information retrieval (ISMIR)
34. Jiang Z, Dannenberg RB (2016) Melody identification in standard midi files. In: Proceedings of the 16th sound & music computing conference, pp 65–71
35. Li L, Junwei C, Lei W, Yan M (2008) Melody extraction from polyphonic midi files based on melody similarity. In: International symposium on information science and engineering, vol. 2. IEEE, pp 232–235
36. Adiloglu K, Noll T, Obermayer K (2006) A paradigmatic approach to extract the melodic structure of a musical piece. J New Music Res 35(3):221–236
37. Zhao W, Zhou Y, Tie Y, Zhao Y (2018) Recurrent neural network for midi music emotion classification. In: IEEE 3rd advanced information technology, electronic and automation control conference (IAEAC). IEEE, pp 2596–2600
38. Conklin D (2006) Melodic analysis with segment classes. Mach Learn 65(2):349–360
39. Jin Y, Wang M (2020) Lstm model for single to dual track piano midi file. In: 2020 IEEE 9th global conference on consumer electronics (GCCE). IEEE, pp 29–31
40. Li T, Chan AB, Chun A (2010) Automatic musical pattern feature extraction using convolutional neural network. Genre 10(2010):1x1
41. Zhang W, Chen Z, Yin F, Zhang Q (2018) Melody extraction from polyphonic music using particle filter and dynamic programming. IEEE/ACM Trans Audio Speech Lang Process 26(9):1620–1632
42. Salamon J, Gómez E (2012) Melody extraction from polyphonic music signals using pitch contour characteristics. IEEE Trans Audio Speech Lang Process 20(6):1759–1770
43. Frieler K, Basaran D, Höger F, Crayencour H-C, Peeters G, Dixon S (2019) Don't hide in the frames: Note-and pattern-based evaluation of automated melody extraction algorithms. In: 6th international conference on digital libraries for musicology, pp 25–32
44. Gómez E, Klapuri A, Meudic B (2003) Melody description and extraction in the context of music content processing. J New Music Res 32(1):23–40
45. Paiva RP, Mendes T, Cardoso A (2006) Melody detection in polyphonic musical signals: exploiting perceptual rules, note salience, and melodic smoothness. Comput Music J 30(4):80–98
46. Lee J, Jang D, Yoon K (2017) Automatic melody extraction algorithm using a convolutional neural network. KSII Trans Internet Inf Syst (TIIS) 11(12):6038–6053
47. Wu R (2021) Research on automatic recognition algorithm of piano music based on convolution neural network. In: Journal of physics: conference series, vol. 1941, no. 1. IOP Publishing, p 012086
48. Choi K, Fazekas G, Sandler M, Cho K (2017) Transfer learning for music classification and regression tasks. arXiv preprint [arXiv:1703.09179](https://arxiv.org/abs/1703.09179)
49. Salamon J, Gómez E, Ellis DP, Richard G (2014) Melody extraction from polyphonic music signals: approaches, applications, and challenges. IEEE Signal Process Mag 31(2):118–134
50. Bittner RM, McFee B, Salamon J, Li P, Bello JP (2017) Deep salience representations for f0 estimation in polyphonic music. In: ISMIR, pp 63–70
51. Ellis DP, Poliner GE (2006) Classification-based melody transcription. Mach Learn 65(2):439–456
52. Bittner RM, Salamon J, Tierney M, Mauch M, Cannam C, Bello JP (2014) Medleydb: a multitrack dataset for annotation-intensive mir research. ISMIR 14:155–160
53. Hsiao W-Y, Liu J-Y, Yeh Y-C, Yang Y-H (2021) Compound word transformer: Learning to compose full-song music over dynamic directed hypergraphs. arXiv preprint [arXiv:2101.02402](https://arxiv.org/abs/2101.02402)
54. Huang Y-S, Yang Y-H (2020) Pop music transformer: beat-based modeling and generation of expressive pop piano compositions. In: Proceedings of the 28th ACM international conference on multimedia, pp 1180–1188
55. Yang Z, Dai Z, Yang Y, Carbonell J, Salakhutdinov RR, Le QV (2019) Xlnet: generalized autoregressive pretraining for language understanding. Adv Neural Inf Process Syst 32
56. Dai Z, Yang Z, Yang Y, Carbonell J, Le QV, Salakhutdinov R (2019) Transformer-xl: attentive language models beyond a fixed-length context. arXiv preprint [arXiv:1901.02860](https://arxiv.org/abs/1901.02860)
57. Chuan C-H, Agres K, Herremans D (2020) From context to concept: exploring semantic relationships in music with word2vec. Neural Comput Appl 32(4):1023–1036
58. Matsunaga R, Abe J-I (2005) Cues for key perception of a melody: pitch set alone? Music Percept 23(2):153–164
59. Hadjeres G, Nielsen F (2020) Anticipation-rnn: enforcing unary constraints in sequence generation, with application to interactive music generation. Neural Comput Appl 32(4):995–1005

60. Ju Z, Lu P, Tan X, Wang R, Zhang C, Wu S, Zhang K, Li X, Qin T, Liu T-Y (2021) Telemelody: lyric-to-melody generation with a template-based two-stage method. arXiv preprint [arXiv:2109.09617](https://arxiv.org/abs/2109.09617)
61. He T, Liu W, Gong C, Yan J, Zhang N (2021) Music plagiarism detection via bipartite graph matching. arXiv preprint [arXiv:2107.09889](https://arxiv.org/abs/2107.09889)
62. Li M, Sleep R (2004) Melody classification using a similarity metric based on kolmogorov complexity. In: Journées d'informatique musicale
63. Bukhsh ZA, Jansen N, Saeed A (2021) Damage detection using in-domain and cross-domain transfer learning. *Neural Comput Appl* 33(24):16921–16936
64. Wu A, Han Y, Zhu L, Yang Y (2021) Universal-prototype enhancing for few-shot object detection. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 9567–9576
65. Ren S, He K, R.Girshick K, Sun J (2015) Faster r-cnn: towards real-time object detection with region proposal networks. *Adv Neural Inf Process Syst* 28
66. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 770–778
67. Wu A, Han Y, Zhu L, Yang Y (2021) Instance-invariant domain adaptive object detection via progressive disentanglement. *IEEE Trans Pattern Anal Mach Intell* 44(8):4178–4193
68. Huang C-ZA, Vaswani A, Uszkoreit J, Shazeer N, Simon I, Hawthorne C, Dai AM, Hoffman MD, Dinculescu M, Eck D (2018) Music transformer. arXiv preprint [arXiv:1809.04281](https://arxiv.org/abs/1809.04281)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.