Fine-Grained Position Helps Memorizing More, a Novel Music Compound Transformer Model with Feature Interaction Fusion

Zuchao Li, Ruhan Gong, Yineng Chen, and Kehua Su*

School of Computer Science, Wuhan University, Wuhan 430072, P. R. China
{zcli-charlie, ruhjgs, chernyn, skh}@whu.edu.cn

Abstract

Due to the particularity of the simultaneous occurrence of multiple events in music sequences, compound Transformer is proposed to deal with the challenge of long sequences. However, there are two deficiencies in the compound Transformer. First, since the order of events is more important for music than natural language, the information provided by the original absolute position embedding is not precise enough. Second, there is an important correlation between the tokens in the compound word, which is ignored by the current compound Transformer. Therefore, in this work, we propose an improved compound Transformer model for music understanding. Specifically, we propose an attribute embedding fusion module and a novel position encoding scheme with absolute-relative consideration. In the attribute embedding fusion module, different attributes are fused through feature permutation by using a multi-head self-attention mechanism in order to capture rich interactions between attributes. In the novel position encoding scheme, we propose RoAR position encoding, which realizes rotational absolute position encoding, relative position encoding, and absolute-relative position interactive encoding, providing clear and rich orders for musical events. Empirical study on four typical music understanding tasks shows that our attribute fusion approach and RoAR position encoding brings large performance gains. In addition, we further investigate the impact of masked language modeling and casual language modeling pre-training on music understanding.

Introduction

Music, an important art form that permeates human's daily life, is usually expressed with symbols for preserving. Like natural language, symbolic music is also composed of a series of symbols associated with each other according to certain rules, which makes music artificial intelligence adopt using similar techniques as natural language. A growing database of music over the past years have provided the conditions for the researches of music understanding including music information retrieval (Casey et al. 2008), music generation (Huang et al. 2019; Sheng et al. 2021), etc. The purpose of music information retrieval is to retrieve useful information from music and categorize it based on music understanding. The most common tasks in music information retrieval are genre classification, velocity prediction, composer classification, emotion classification, and so on. Rather, music generation generates musical symbol sequences based on certain requirements, such as emotion, theme, and prompt. It is said that what cannot be understood cannot be made, so musical understanding is also crucial to generation.

Recent studies have demonstrated the success of pre-trained models (e.g., BERT) in handling natural language processing tasks and learning general language representations from unlabeled texts. Due to the great success of pre-trained models using Transformers (Vaswani et al. 2017), many attempts have been made to introduce the Transformer-like architectures to symbolic music understanding tasks. Unlike tokens in natural language processing, musical sequences in symbolic music usually involve tokens of various types such as pitch, duration, and velocity. There are two typical represetations: REMI (Huang and Yang 2020) and CP (Hsiao et al. 2021). In REMI representation, different token types can be placed in the same sequence, making the symbolic music exactly the same as the natural language for input. While in CP representation, different types of tokens are placed in different sequences for short lengths. In other words, multiple token types are grouped at a single position in the input sequence, resulting in the symbolic music having inputs with multiple subsequences.

Although the Compound Word Transformer (CP+Transformer) has achieved performance improvements on music understanding, the grouping mechanism of CP and the characteristics of music symbols still leave room for improvement. On the one hand, there is a lack of interaction between the various types of token features after grouping. This does not exist in the REMI representation, since various types of tokens are placed in the same sequence and interacted when encoding. Since a music event is composed of multiple related attributes, like structural (e.g., bar, sub-beat) and diverse information (e.g., tempo, and pitch), it is unwise to separate event attribute tokens. On the other hand, the difficulty in understanding music sequences is not only caused by the sequence being too long. This is because music sequences are more sensitive

*Corresponding author. This work is supported by the National Natural Science Foundation of China under Grant 62272354.

to position information than natural language sequences. Empirically, disrupting a few orders of natural language may not have much effect on the overall meaning, while a few disorders of events for music can even make the music harsh or even noisy. Therefore, it is not sufficient for music understanding to rely on the previous absolute position or relative position encoding alone.

To address the two shortcomings, we propose a feature interaction fusion (FiF) module and a rotational absolute-relative (RoAR) position encoding method. Specifically, the FiF module implements the intra-group token feature interaction of compound words to obtain better input feature representation. And in RoAR position encoding, the consideration of absolute position, relative position, and absolute-relative interaction is introduced to improve the Transformer model's awareness of the position information for memorizing more effective features. We conduct experiments on four symbolic music comprehension benchmarks: *melody*, *velocity*, *composer*, and *emotion* classification. The resultant models obtain state-of-the-new performance in 97.59, 53.73, 80.95, 76.15, respectively with large margins compared to the baseline. Analysis fully shows the effectiveness of our proposed method in music understanding. And we set a new powerful music understanding model[1] for the music AI community.

Related Work

Music Symbolic Representation

Music is processed into symbols for understanding and recording. Representation forms typically consist of MusicXML, MIDI, REMI, CP, etc. MIDI (Oore et al. 2020) is a symbolic representation which encodes each note with a "note-on" token, a "note-off" token, and a "velocity" token. The "note-on" token and "note-off" token represent the start and release of a note respectively, and the "velocity" token changes the velocity applied to all subsequent notes. Besides, a "time-shift" token is used to move the time step forward by millisecond. REMI (Huang and Yang 2020) uses a beat-based manner to represent the advancement in time. A "bar" token indicates the start of a new bar, and a "sub-beat" token points to one of the discrete sub-beat divisions in a bar. Moreover, REMI uses "note-duration" in replacement of "note-off" and generates tempo and chord explicitly to make them controllable. Based on REMI, Hsiao et al. (2021) distinguishes different types of tokens and creatively groups tokens belonging to the same family into "compound words" (CP). Since the compound words compress multiple tokens into a super token, it needs much less time steps to represent the same music piece compared with REMI.

Based on these symbolic representations, Simonetta et al. (2019) propose a convolutional neural network (CNN)-based model which represents MIDI with piano roll representation (Dong, Hsiao, and Yang 2018) for melody identification. Meanwhile, Jiang and Dannenberg (2019) present a Bayesian probability model to identify the melody in standard MIDI files. Attempts have been made by Kim et al. (2020) and Kong, Choi, and Wang (2020) to use CNN-based deep neural network classifiers for composer classification in the piano roll (Dong, Hsiao, and Yang 2018) MIDI format. Hung et al. (2021) present prototypes of models using RNN-based classifier called "BiLSTM-Attn" (Lin et al. 2017) for emotion classification in the format of REMI and CP.

[1]Code will be available at https://github.com/zcli-charlie/MIDI-FiF-RoAR.

Music Pre-training

Tsai and Ji (2020) firstly leveraged pre-trained Transformer for symbolic music classification, demonstrating that Transformer outperforms CNN (Verma and Thickstun 2019) and LSTM (Merity, Keskar, and Socher 2018) models on a 9-way composer style classification task. Zeng et al. (2021) also developed a large-scale symbolic music understanding dataset that covers more than a million songs and pre-trained MusicBERT with this large dataset. To enhance the effect of pre-training, MusicBERT proposed a new OctupleMIDI encoding and bar-level masking strategy. Soon after MusicBERT, Chou et al. (2021) presented MidiBERT-Piano, a large-scale pre-trained model, which employs the masked language modeling (Devlin et al. 2019) to pre-train a 12-layer Transformer model. Results show that MidiBERT-Piano outperforms RNN-based baselines with large margins. Li et al. (2022) proposed a fuzzy training framework for controllable music-to-lyric generation.

Symbolic music differs from natural language in several ways. In symbolic music, there are multiple features for the basic elements, and there are multiple tracks for music, which means not only the sequence associations within the track must be considered, but also the relationships between the tracks. Consequently, Miao and Yang (2021) proposed a contrastive learning method for Musical Representation with NonupleMIDI (NonupleCLMR), which incorporates NonupleMIDI encoding and a simCLMR mechanism for musical representations learning. To overcome the shortcoming of ignoring the intrinsic structure of music in the masked language model pre-training (Sun, Li, and Zhao 2021), Qiu, Chen, and Zhang (2022) proposed a Symbolic Music Emotion Recognition (SMER) multi-task pre-training framework. In conjunction with the emotion recognition-assisted intrinsic structure pre-training approaches, the experiments demonstrated better emotion recognition performance.

Position Encoding

Compared to CNNs and RNNs, Transformers achieve generally better performance, however, unlike CNN and RNN, which are inherently encoded, Transformer needs to apply position encoding due to its property of permutation equivalent. Recently, many explorations have been put into the position encoding, which can be categorized into following groups:

Absolute Position Encoding One way to add absolute position encoding is through simply adding it to the input elements and train it through the training process, which suffers the criticism for its lack of ability to extrapolate (Gehring et al. 2017). Vaswani et al. (2017) proposed absolute position encodings in a sine and cosine function form of different frequencies, and thus each dimension of them corresponds

to a sinusoid. However, there are some problems with sinusoidal encoding and embedding layer. Liu et al. (2020) suggested that the former is manually designed, containing no learnable parameters and the latter is limited to the max length of input sequences. To tackle this issue, borrowing the idea from ODE approach, Liu et al. (2020) proposes a new position layer which is learnable and aims to achieve higher flexibility with respect to different set.

Relative Position Encoding Shaw, Uszkoreit, and Vaswani (2018)'s work was pioneering in proposing relative position encoding in Transformer model, which performs better than many absolute counterparts. In order to reuse the hidden states while keeping position information, Dai et al. (2019) proposes Transformer-XL. One innovation point is that in this method, position information is injected into the attention score of each layer instead of initial embeddings. According to He et al. (2021), the computation of the attention weights in the methods discussed above are limited to content-to-content and content-to-position terms. They further emphasize the importance of the position-to-content term and propose a novel relative position encoding in DeBERTa (He et al. 2021).

Other Position Encoding Wang et al. (2020a) extend CNN, RNN and Transformer to complex space. Their complex-style encoding is competent to model both the global absolute positions of tokens along with order relationships. Su et al. (2021) also leverages complex analysis, but it's just used for derivation. In fact, it is a rotary-style encoding which transforms absolute position information into relative position information.

Feature Interaction

Feature interaction methods are often used for embedding fusion in knowledge graphs to improve feature completeness. To combine entity and relation embeddings, Socher et al. (2013) proposed Neural Tensor Networks. There are several studies (Dong et al. 2014; Ravishankar, Dewangan, and Talukdar 2017) that use a Multi-Layer Perceptron to score entities and relations. To calculate the score vector, Dettmers et al. (2018) proposed ConvE, which uses convolutional filters over reshaped subject and relation embeddings. A variant of ConvE, ConvTransE, was proposed by Shang et al. (2019), which directly applies convolution to the layered subject and relation embeddings instead of using 2D reshaping. Using ConvKB (Nguyen et al. 2018), convolutional filters of width 1 are applied to the stacked subject, relation, and object embeddings for feature interaction. Vashishth et al. (2020) proposed InteractE, which is based on three key ideas – feature permutation, feature reshaping, and circular convolution.

Methodology

For music understanding tasks, musical melody is used as input, and MIDI is the most common format which can be considered as a sequence of musical events to describe it. Following the practice of Huang and Yang (2020); Hsiao et al. (2021), we transform the MIDI scores into an event token sequence, thus transforming music understanding into similar tasks as natural language understanding. Recently, Transformer-based pre-training–fine-tuning paradigm has become the dominant approach in natural language processing, and thus in this work we propose new modules based on Transformer architecture to improve the music understanding.

Background

There are currently two main formats of MIDI-converted token sequences: Revamped MIDI-derived events (REMI) (Huang and Yang 2020) and Compound Word (CP) (Hsiao et al. 2021), where CP takes advantage of the simultaneous occurrence of multiple musical events to alleviate the issue of long sequences in the REMI format. Due to the fact that CP is the main format studied in this work, we describe our method formally using CP. Formally, given a sequence of MIDI-transformed CPs as $S = \{[t^{\text{bar-beat}}, t^{\text{sub-beat}}, t^{\text{pitch}}, t^{\text{duration}}]_i, 1 \leq i \leq N\}$, where N is the sequence length, $t^{\text{bar-beat}}$, $t^{\text{sub-beat}}$, t^{pitch}, and t^{duration} are four types of event tokens that are compounded into a single word. In the original CP paper (Hsiao et al. 2021), a total of eight token types were obtained, but MIDI in music understanding contains two main categories of events, MIDI scores and MIDI performance. The former is a faithful record of musical properties, while the latter indicates attributes of performance. We follow the practice of Chou et al. (2021) and only use MIDI scores for music understanding, and discard tokens related to the performance aspects of music, such as note velocity and tempo.

After obtaining the compound word sequence S, multiple-embedding mapping and concatenation is adopted to transform the tokens into a joint token representation:

$$\begin{aligned} \mathbf{e}_i^{A_j} &= \text{Emb}^{A_j}(t_i^{A_j}), \\ \mathbf{X}_i &= \text{Linear}([\mathbf{e}_i^{A_1} \oplus \mathbf{e}_i^{A_2} \oplus ... \oplus \mathbf{e}_i^{A_J}]), \end{aligned} \tag{1}$$

where $A = \{\text{bar-beat}, \text{sub-beat}, \text{pitch}, \text{duration}\}$ with size J, and $\oplus$ indicate the concatenation process. The vectorized token representation $\mathbf{W}$ is then further fed into the Transformer encoder for its contextualized representation $\mathbf{H}$. In line with language words, MIDI event tokens are also sequential, so position vectors are adopted as indicators of sequential order. Since MIDI token sequences are relatively long, they are usually encoded with relative positions (Ke, He, and Liu 2021), and the final encoding process in the Transformer is as follows. For the k-th Transformer layer, its input is written as $\mathbf{H}_{k-1}$ and the encoded output is $\mathbf{H}_k$, where $\mathbf{H}_0 = X$.

$$Q = \mathbf{W}^Q \mathbf{H}_{k-1}, K = \mathbf{W}^K \mathbf{H}_{k-1}, V = \mathbf{W}^V \mathbf{H}_{k-1},$$

$$\mathbf{H}_k = \text{MSA}(\mathbf{H}_{k-1}) = \textsc{Concat}(\textsc{Softmax}(\frac{Q_h K_h^T}{\sqrt{d_h}})V_h)\mathbf{W}^O,$$

where Q_h, K_h, V_h indicates sub-chunk representation for head h respectively. Relative position is incorporated into the calculation of attention weights, which are previous calculated by $Q_h K_h^T$, by introducing relation position indication terms. Ke, He, and Liu (2021) directly model the relationships between a pair of positions using different projection

matrices:

$$\mathbf{a}_{i,j} = \mathbf{q}_i \mathbf{k}_j^T + (\mathbf{p}_i \mathbf{U}^Q)(\mathbf{p}_j \mathbf{U}^K)^T, \quad (1)$$

where $a_{i,j}$ represent the attention weights between position i and j, $\mathbf{p}_i$ and $\mathbf{p}_j$ represent the position embedding in positions i and j respectively, and $\mathbf{U}^Q$ and $\mathbf{U}^K$ are the projection weights for the position embeddings.

Feature Interaction Fusion

In Compound Word Transformer, embedding representations of different type of tokens are concatenated and then linear mapped for eigenspace transformation. Since CP represents multiple simultaneous musical events, there are associations between different tokens inside a word, but the current embedding fusion process ignores them. Furthermore, in the subsequent Transformer encoding, the attention mechanism is applied to the sequence dimension rather than the feature dimension of each token, so the features of these tokens also do not interact with each other. It is harmful for music understanding due to music is understood on the basis of multiple events occurring simultaneously rather than individual events.

For this characteristic, we propose a feature interaction fusion module, which acts before the embedding vector enters the Transformer encoding. Specifically, we first perform a multi-head self-attention operation across the token embeddings of each compound word to capture the relationship between the tokens within the compound word instead of directly concatenating them:

$$\begin{aligned} \mathbf{E}^A &= [\mathbf{E}^{A_1}, \mathbf{E}^{A_2}, ..., \mathbf{E}^{A_J}], \\ \mathcal{G} &= \textsc{Softmax}(\mathbf{W}_A^Q \mathbf{E}^A (\mathbf{W}_A^K \mathbf{E}^A)^T), \end{aligned} \quad (2)$$

where $\mathbf{E}^A$ indicates that embeddings of different event tokens are placed in separate tensor. To facilitate the analysis, we write a simplified form of multi-head self-attention, ignoring the multi-head mechanism and scaling factors. Multi-head self-attention computes attention weights $\mathcal{G}$ across event tokens to find the association between event types.

After obtaining the association between event tokens, we perform feature interaction operation on the embeddings of different event token according to the weights $\mathcal{G}$, which actually leads to that the representations of each event token are a weighted sum of the features on all types of event tokens:

$$\hat{\mathbf{X}}_i = \sum_{j=1}^{J} \mathcal{G}_{i,j} \mathbf{e}_i^{A_j}. \quad (3)$$

Based on the output of feature interaction, we further employ a linear projection layer to project the interacted features into a uniform feature space for the next input:

$$\mathbf{X}_i = \text{Linear}(\hat{\mathbf{X}}_i). \quad (4)$$

RoAR Position Encoding

Since position information is more important for music understanding than natural language, we further propose rotational absolute-relative position encoding which allows considering both absolute position information and relative position information. There are two types of position information currently available: absolute and relative. An absolute position is indicated by adding additional positioning representations or applying additional transformations to the inputs. Generally, there are two ways to applying absolute position encoding – rotating the input to represent the position with angle information or superimposing position feature on the input. Since our token embedding features come from the fusion of various embeddings, we choose the rotary implementation since in this way the norm of the feature vector will not be changed after applying absolute position encoding.

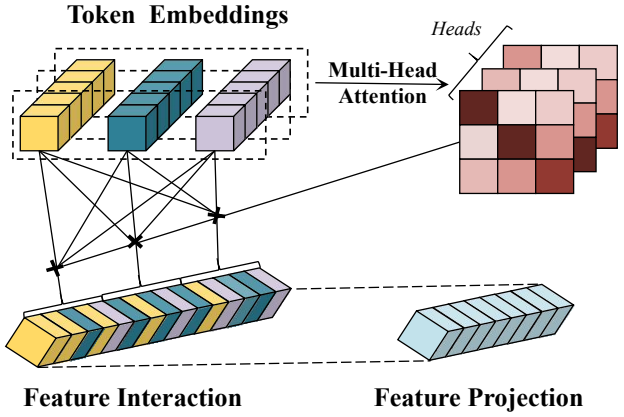

Figure 1: Overview of attribute embedding fusion module. Different color in token embeddings is to show embeddings of different attributes. Color in feature interaction is to illustrate that FiF fused different attributes into one compound word according to their relationship. Different color in attention map is to show the relationship of different attributes. And color in feature projection means the attentional representation is mapped into a new feature space. "x" is to show the multiplication operation between the token embeddings and attention map.

Motivated by rotary position encoding (RoPE) in Su et al. (2021), we denote $f(Z, m) = Z \circ e^{im\theta}$ as applying a rotation operation to the vector Z according to the absolute position m, where e is the base of natural logarithms and $\circ e^{im\theta}$ is a special mark used to express the applied rotation operation. Notably, $e^{im\theta}$ does not represent a complex number here, instead, it just borrows the concept of complex analysis to express the rotation of a vector. In fact, the practical computation process of $\circ e^{im\theta}$ is:

$$\begin{pmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \\ \vdots \\ z_{d-1} \\ z_d \end{pmatrix} \otimes \begin{pmatrix} \cos m\theta_1 \\ \cos m\theta_1 \\ \cos m\theta_2 \\ \cos m\theta_2 \\ \vdots \\ \cos m\theta_{d/2} \\ \cos m\theta_{d/2} \end{pmatrix} + \begin{pmatrix} -z_2 \\ z_1 \\ -z_4 \\ z_3 \\ \vdots \\ -z_d \\ z_{d-1} \end{pmatrix} \otimes \begin{pmatrix} \sin m\theta_1 \\ \sin m\theta_1 \\ \sin m\theta_2 \\ \sin m\theta_2 \\ \vdots \\ \sin m\theta_{d/2} \\ \sin m\theta_{d/2} \end{pmatrix},$$

where d is the dimension of Z, θ is a rotation angle containing absolute position information, and $\otimes$ denotes the element-wise multiplication. In multi-head self-attention, we get the counterparts of Q and K which bear the absolute

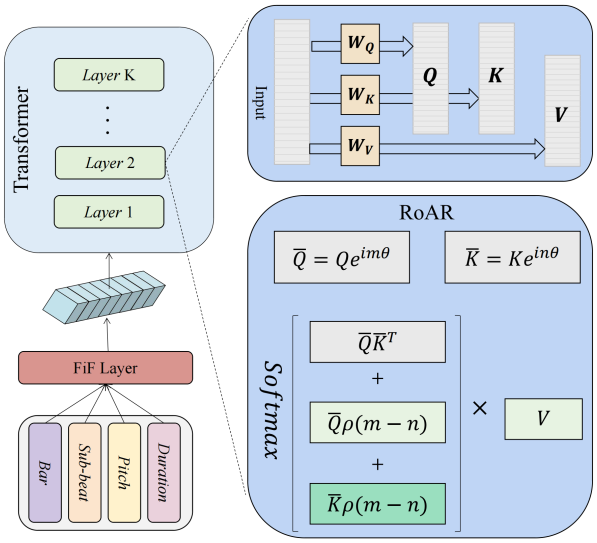

Figure 2: Overview of our proposed model.

position information by $\bar{Q} = Q \circ e^{im\theta}$ and $\bar{K} = K \circ e^{in\theta}$.

In multi-head self-attention, inner product of $\bar{Q}$ and $\bar{K}$ is performed for attention weights. For the convenience of derivation, we put the rotation operation back into the complex space:

$$\begin{aligned}\langle f(Q,m), f(K,n)\rangle &= \langle \bar{Q}, \bar{K}\rangle = \bar{Q}\bar{K}^T \\ &= Qe^{im\theta}K^T e^{i(-n)\theta} = QK^T e^{i(m-n)\theta}.\end{aligned} \tag{5}$$

In this way, the absolute position encoding implemented with rotational operation is completely transformed into relative position encoding. In other words, it actually loses the information of absolute position. Therefore, we argue that RoPE is not sufficient for music understanding since position information is very important for musical sequence.

To tackle this issue, we introduce two additional position terms, and propose a novel RoAR position encoding. The final attention calculation with RoAR is:

$$\bar{Q}\bar{K}^T + \bar{Q}\rho^T_{m-n} + \bar{K}\rho^T_{m-n}, \tag{6}$$

where the first term introduces relative position information as in the original RoPE, the second term and third term introduce relative position information to the input vector with absolute information $\bar{Q}$ and $\bar{K}$ respectively, ρ_{m-n} is a learnable relative position embedding for relative position $m-n$. To our best knowledge, our RoAR is the first position encoding scheme that combines absolute position and relative position.

Discussion Although according to Su et al. (2021), RoPE realizes relative position encoding by means of absolute position encoding, we now show that the relative information in RoPE is not sufficient and we provide a intuitive analysis to explain the gap between it and our RoAR. First of all, observe the equation proposed in original RoPE:

$$q_m k_n^T e^{i(m-n)\theta}, \tag{7}$$

where q_m and k_n denote the m-th and n-th component of queries and keys in attention respectively. If we pick the $(m+l)$-th and $(n+l)$-th component, then the equation is:

$$q_{m+l}k^T_{n+l}e^{i(m-n)\theta}. \tag{8}$$

Suppose that q_{m+l} is quite similar to q_m and k_{n+l} is quite similar to k_n, then the difference between these two equations are so small that it may be hard to be distinguished, which may lead to confusion in encoding. Notably, in natural language processing, since the vocabulary is large (e.g. 20000+), this situation rarely occurs, while for music symbols, the vocabulary is small (e.g. 0-100), so this situation cannot be neglected. Therefore, a new position encoding scheme is needed for music symbolic understanding.

While for our RoAR, we show that it is able to address this problem by introducing two position terms which incorporate absolute-relative position information:

$$q_m k_n^T e^{i(m-n)\theta} + q_m e^{im\theta}\rho^T_{m-n} + k_n e^{in\theta}\rho^T_{m-n}. \tag{9}$$

We argue that the additional position terms play a significant role. To examine this assumption, we only analyze the first term since the two terms can be illustrated in a similar form.

Specifically, The first term corresponding to q_m and k_n is:

$$q_m e^{im\theta}\rho^T_{m-n}. \tag{10}$$

And the first term corresponding to q_{m+l} and k_{n+l} is:

$$q_{m+l}e^{i(m+l)\theta}\rho^T_{m-n} = q_{m+l}e^{im\theta}e^{il\theta}\rho^T_{m-n}. \tag{11}$$

When comparing the two terms, even if q_m may be similar to q_{m+l} in value, the extra rotation $e^{il\theta}$ can distinguish them. Note that the rotation angle will not exceed 2π according to our hyper-parameters, which means there is no need to worry about problems caused by periodicity. Thus, we show that our RoAR is more effective than RoPE for music symbolic understanding tasks.

Pre-training Approach

MLM As in natural language processing, most music understanding pre-training utilize masked language modeling (MLM), which is first proposed in BERT (Devlin et al. 2019) for learning bidirectional representations on unlabeled corpus. In our music understanding pre-training, MLM is also adopted. For a CP sequence S with the length of N, MLM randomly masks tokens at some positions with a special symbol `[MASK]`, and asks the Transformer encoder to predict the masked tokens based on the visible tokens. Denote $\mathcal{M}$ as the set of masked positions, $S_{\mathcal{M}}$ as the set of masked tokens, and $S_{\backslash\mathcal{M}}$ as the masked sequence. The model θ is pre-trained by maximizing the following objective:

$$\log P(S_{\mathcal{M}}|S_{\backslash\mathcal{M}};\theta) \approx \sum_{S_i\in\mathcal{M}} \log P(S_i|S_{\backslash\mathcal{M}};\theta). \tag{12}$$

CLM In natural language, there is another common pre-training approach, represented by GPT (Radford et al. 2018), that relies on unidirectional masking for next token prediction: casual language modeling (CLM). To investigate the effect of this generative pre-training on music understanding, we also involve this pre-training approach. For

a CP sequence S, CLM predicts the probability of tokens $S_i = [t_i^{\text{bar-beat}}, t_i^{\text{sub-beat}}, t_i^{\text{pitch}}, t_i^{\text{duration}}]$ given the previous tokens in the sequence $S_{<i}$: $P(S_i|S_{<i}, \theta)$. The model iteratively predicts the token at all positions by maximizing the following objective:

$$\log P(S;\theta) = \sum_{i=1}^{N} \log P(S_i|S_{<i};\theta), \tag{13}$$

where $S_0 = [\texttt{BOS}^{\text{bar-beat}}, \texttt{BOS}^{\text{sub-beat}}, \texttt{BOS}^{\text{pitch}}, \texttt{BOS}^{\text{duration}}]$.

Experiments

Setup

Datasets We leverage five datasets for pre-training and finetuning: **Pop1K7**, **ASAP**, **POP909**, **Pianist8**, and **EMOPIA**. The information of the five datasets we use is summarized in table 2. All the pieces of five datasets are converted into the representation of MIDI scores (Oore et al. 2020) and are in 4/4 time signature (four beats per bar). Our data setup remains exactly the same as our baseline, MidiBERT (Chou et al. 2021).

- **Pop1K7** (Hsiao et al. 2021) comprises about 108 hours machine transcriptions of 1,748 pop piano recordings.[2] The preprocess is done with a sequence of operations including transcription, synchronization, and quantization. Transcription estimates the pitch, onetime and offset time of the musical notes with the "Onset and Frames" RNN automatic piano transcription model (Hawthorne et al. 2018). Synchronization estimates the downbeat and the beat positions using the RNN-based model from the Madmom library (Böck et al. 2016). Quantization quantizes the onset time and duration of the notes to reduce the size of the vocabulary.
- **ASAP** (Foscarin et al. 2020) is composed of 222 digital musical scores and 1068 MIDI performances of Western classical piano music from 15 different composers.[3]
- **POP909** (Wang et al. 2020b) contains piano of 909 popular songs created by professional musicians.[4] Since POP909 is a MIDI performance dataset containing velocity information, it is used in both melody extraction and velocity prediction tasks.
- **Pianist8** (Chou et al. 2021) consists of 411 pieces original piano music performed by 8 composers.[5] Each is paired with corresponding MIDI performance transcribed by the piano transcription model proposed by Kong et al. (Kong et al. 2020). We use the dataset for composer classification task.
- **EMOPIA** (Hung et al. 2021) includes 1,087 pop piano music clips from 387 songs and clip-level emotion labels.[6] The dataset is used for emotion classification task.

[2]https://github.com/YatingMusic/compound-word-transformer
[3]https://github.com/fosfrancesco/asap-dataset
[4]https://github.com/music-x-lab/POP909-Dataset
[5]https://zenodo.org/record/5089279
[6]https://annahung31.github.io/EMOPIA/

Training Details Following Chou et al. (2021), we adopt $\text{BERT}_{\text{base}}$ (Devlin et al. 2019) as the model backbone, which has 12 layers of multi-head self-attention with 12 heads for each, the hidden size of the model is 768, and the total parameters of the model is 110M. MLM and CLM are combined to pre-train the model by cross-iteratively updating the parameters in one training process (i.e., MLM-CLM-MLM-CLM-...). In our implementation, MLM and CLM use the same linear projection output layer in token reconstruction since the target shares same vocabulary, but with the different first token. The training target of MLM are like "[MLM], token_1, [mask], token_3, ..., [EOS]", while for CLM is like "[CLM], token_1, token_2, ..., [EOS]". Particularly, it only needs to predict one token at a time for REMI, while four tokens for CP.

We use 85% of the given corpus for pre-training, and 15% for validation. We pre-train the model using the CP representation on 2 GeForce RTX 3090 GPUs for about 2 days on average for three different position encodings. While pre-training using REMI under the same settings takes at least 5 days. We use a batch size of 16 for CP and 20 for REMI when pre-training. The model uses AdamW optimizer (Loshchilov and Hutter 2019) with learning rate 2e$-$5 and weight decay rate 0.01, and the max sequence length is set to 512. We stop the pre-training which has at most 500 epochs if the validation accuracy does not improve for 30 consecutive epochs.

The ratio of training, validation and test splits for the four downstream tasks is set to 8:1:1. The batch size of both CP and REMI representation is uniformly set to 12 for fine-tuning. Like pre-training, we fine-tune the model for at most 10 epochs on a GeForce RTX 3090 GPU for less than half an hour. If there is no improvement for three continuous epochs, we stop the fine-tune process in advance. The same parameters are used to initialize the models for four downstream tasks.

Results Analysis

In Table 1, we show the performance of two typical token representations REMI and CP on the four benchmarks. In addition, we also list the performance of traditional RNN model as well as the previous state-of-the-art pre-trained symbolic music understanding models, MidiGPT, MidiBERT, etc. First, comparing MidiBERT with our reproduced MidiBERT†, we found that MidiBERT† has basically the same performance as MidiBERT on Melody and Velocity tasks. On Composor tasks, the reproduced MidiBERT† obtained generally better results, and on the Emotion task, the effect is slightly reduced on the REMI representation, while on the CP representation we obtain better performance. This demonstrated that our reproduced MidiBERT† is a strong baseline. Further comparing the baseline MidiBERT† with our full model, we found improvements of 0.93, 2.09, 4.99, 7.83 (REMI), 1.09, 2.03, 1.58, 6.43 (CP) on the four tasks, respectively. This suggests that our feature interaction fusion and RoAR position encoding bring powerful gains for symbolic music understanding. The largest improvement comes from the Emotion task, which indicates that the fine-grained position encoding and the co-relationship between musical

Systems	Token	**Melody**	**Velocity**	**Composer**	**Emotion**	Training Hours
MidiGPT (Ferreira, Lelis, and Whitehead 2020)	CP	–	–	–	61.88	– / –
RNN (Chou et al. 2021)	REMI	89.96	44.56	51.97	53.46	– / –
	CP	88.66	43.77	60.32	54.13	– / –
MidiBERT (Chou et al. 2021)	REMI	90.97	49.02	67.19	67.74	– / –
	CP	96.37	51.63	78.57	67.89	– / –
MT-MidiGPT (Qiu, Chen, and Zhang 2022)	CP	–	–	–	66.95	– / –
MT-MidiBERT (Qiu, Chen, and Zhang 2022)	CP	–	–	–	69.97	– / –
MidiBERT†	REMI	90.91	49.31	73.49	65.90	119.5h / 18.7min
	CP	96.50	51.70	79.37	69.72	48.2h / 11.7min
Ours	REMI	**91.84** (↑ **0.93**)	**51.40** (↑ **2.09**)	**78.48** (↑ **4.99**)	**73.73** (↑ **7.83**)	120.3h / 22.0min
	CP	**97.59** (↑ **1.09**)	**53.73** (↑ **2.03**)	**80.95** (↑ **1.58**)	**76.15** (↑ **6.43**)	50.1h / 11.5min

Table 1: Performance of our model on four benchmarks. MdiBERT† indicates the results of our own reproduction.

Dataset	Pieces	Hours	AvgND	AvgBPP
Pre-training				
Pop1K7	1,748	108.8	8.5	103.3
$ASAP_{4/4}$	65	3.5	2.9	95.9
Pre-training & Fine-tuning				
(*Task: Melody/Velocity*)				
$POP909_{4/4}$	865	59.7	6.1	94.9
(*Task: Composer*)				
Pianist8	411	31.9	9.6	108.9
(*Task: Emotion*)				
EMOPIA	1,078	12.0	10.0	14.8

Table 2: Datasets statistics.

	Melody	**Velocity**	**Composer**	**Emotion**
MidiBERT†	96.50	51.70	79.37	69.72
Full Model	97.59	53.73	80.95	76.15
w/o FiF	97.16	53.04	76.19	75.23
w/o RoAR	97.18	53.54	80.95	72.48
w/o Pre-training	93.19	48.67	57.14	59.63
w/o Fine-tuning†	91.72	47.05	75.40	69.72

Table 3: Ablation study for model designs and training. w/o Fine-tuning† means to fix the pre-trained parameters and only update the network of the downstream task.

event attributes are very important for musical emotional understanding. Besides, compare our model with baseline without pre-training (RNN), the performance gap is even larger, which shows that the pre-training strategies of MLM and CLM help the model learn to understand the structure of symbolic music. We also compared the training time of MidiBERT† with our model, and found no significant difference between the two, which shows that our FiF and RoAR are an effective and efficient enhancement for symbolic music understanding. We have noticed some recent works on music pre-training (i.e., MuseBERT (Wang and Xia 2021)), since our model is different from MuseBERT in pre-training datasets and model scale, it's not fair to compare the two models directly. Besides, our contributions focus on structural improvement of symbolic music, which are orthogonal with MuseBERT. We leave the further exploration to future work.

Ablation Study

We present the ablation study for the model design and training methods in Table 3. First, after removing FiF or RoAR, we found that the results have decreased, which shows that both FiF and RoAR have played a role in improving the performance, but they are still higher than the baseline MidiBERT†, indicating that both designs improve the baseline. The most noticeable drops are w/o FiF settings in Composer task and w/o RoAR in Emotion task. This shows that for the Composer task, the relationship between the event attributes is more important, and the fine-grained position information in the Emotion task is more important. For composer classification, the combination of specific attributes is the characteristic of different composers, while in the emotion task, both global information and local information may both affects the emotion understanding. Therefore the interaction of multiple music attributes (FiF) contributes more to the composer task, while fine grained position information (RoAR) is more helpful to the emotion task. Second, in terms of model training, the effect of not using pre-training is greatly reduced, indicating that pre-training is very important for symbolic music understanding. In addition, in the finetune stage, fixing the pre-trained parameters also leads to partially performance decrease, indicating that there is a difference between pre-training and specific music understanding tasks, so it is necessary to continue to update the parameters in the finetune stage.

Further Exploration

Effects of Feature Interaction Fusion To further demonstrate the effectiveness of our feature interaction fusion method for multiple musical event attributes encoding, we also compare two other advanced feature fusion methods, DistMult (Yang et al. 2014) and InteractE (Vashishth et al. 2020), and the results are shown in Table 4. Results show

	Melody	Velocity	Composer	Emotion
Ours	97.59	53.73	80.95	76.15
DistMult	97.32	53.26	78.42	75.69
InteractE	97.25	53.35	79.36	75.88
w/o FiF	97.16	53.04	76.19	75.23

Table 4: Performance of different feature interaction fusion.

	Melody	Velocity	Composer	Emotion
RoAR	97.59	53.73	80.95	76.15
w/o $\bar{K}\rho_{m-n}$	97.25	53.32	79.31	75.61
w/o $\bar{Q}\rho_{m-n}$	97.19	53.44	79.53	74.92
w/o AR	97.10	53.25	78.88	74.06
RoPE	97.08	53.14	77.78	72.48
RelPE	96.55	52.42	79.37	71.56
AbsPE	95.57	51.03	72.22	69.72

Table 5: Performance of different position encoding.

that compared with no feature fusion, the three feature fusion methods all bring about gains in music understanding. Our fusion method outperforms DistMult and InteractE for multi-attribute feature fusion of music events, indicating that the attention based fusion is more suitable for multiple music attributes than convolutional operation.

Effects of Different Position Encoding To investigate the effect of position encoding on symbolic music understanding, we report the effects of different RoAR designs in Table 5, along with the experimental results of common rotational position encoding (RoPE), relative position encoding (RelPE), and absolute position encoding (AbsPE). In RoAR, w/o $\bar{K}\rho_{m-n}$ and w/o $\bar{Q}\rho_{m-n}$ represent the removal of corresponding position term as in Eq. (6), while w/o absolute-relative (AR) means that the mechanism of interaction between relative position and absolute position is removed, and (6) is modified to $\bar{Q}\bar{K}^T + Q\rho_{m-n}^T + K\rho_{m-n}^T$. Results suggest that w/o $\bar{K}\rho_{m-n}$ and w/o $\bar{Q}\rho_{m-n}$ and w/o AR all cause the decrease of the results, among which w/o AR decreases the most, indicating that the mechanism of relative position and absolute position interaction is the most important for the success of RoAR. Compared with other position encodings, it is found that AbsPE has the worst performance for understanding of symbolic music. RoPE and RelPE are comparable, but they are all lower than our proposed RoAR, indicating that RoAR is a fine-grained position encoding, which is more suitable for symbolic music understanding.

Effects of Data Scale In addition to the structure, the influence of the pre-training data scale on the results has also attracted researcher's attention. We compare the performance of MidiBERT and our model on different pre-training dataset sizes in Table 6. The results show that the reduction of pre-training data will lead to the decline of symbolic music understanding ability, but compared with MidiBERT, our model has a relative smaller decline, which shows that the effective understanding structure design, FiF and RoAR, can help reduce the requirements on the size of pre-training data .

Model	Data Size	Melody	Velocity	Composer	Emotion
MidiBERT	4,167	96.37	51.63	78.57	67.89
Ours		97.59	53.73	80.95	76.15
MidiBERT	3,696	96.15	52.11	67.46	64.22
Ours		96.91	52.59	72.22	66.97
MidiBERT	1,813	95.35	48.73	58.73	67.89
Ours		96.52	50.73	65.87	69.72

Table 6: Performance of different pre-training data size.

	Melody	Velocity	Composer	Emotion
MLM+CLM	97.59	53.73	80.95	76.15
MLM	97.50	52.35	74.60	72.48
CLM	95.77	52.03	79.37	70.64

Table 7: Performance of different pre-training strategy.

MLM vs CLM in Music Understanding As two typical pre-training strategies, MLM and CLM are two lines of battle in NLP, and there is a lack of research on the impact of MLM and CLM on symbolic music understanding. Therefore, we show the performance of Transformer model with CP+FiF+RoAR setting in Table 7 to explore the impact of different pre-training strategies on downstream tasks. The comparison results show that both MLM and CLM alone are not as good as MLM+CLM, which indicates that both bidirectional (MLM) and unidirectional (CLM) understanding are required for symbolic music understanding. From the perspective of different tasks, MLM on Melody, Velocity and Emotion has more advantages, while on Composer, CLM has more advantages, indicating that bidirectional understanding may not be stronger than unidirectional understanding. And the enhanced effect mainly depends on the characteristics of downstream tasks , such as the more consistent characteristics of unidirectional understanding and left-to-right composing music.

Conclusion

In this paper, we propose feature interaction fusion and rotational absolute-relative interaction position encoding according to the characteristics of symbolic music understanding, in order to cope with the multi-attribute encoding of music events and the requirements for fine-grained position information. Experimental results on four symbolic music comprehension benchmarks show that FiF and RoAR position encoding bring a substantial performance improvement over the baseline, reaching a new state-of-the-art. And we have also conducted a careful study of the position encoding scheme and pre-training method for symbolic music understanding, showing the characteristics of symbolic music understanding different from natural language. The proposed system provides a strong baseline for follow-up research in the symbolic music understanding community.

References

Böck, S.; Korzeniowski, F.; Schlüter, J.; Krebs, F.; and Widmer, G. 2016. madmom: A New Python Audio and Music Signal Processing Library. In Hanjalic, A.; Snoek, C.; Worring, M.; Bulterman, D. C. A.; Huet, B.; Kelliher, A.; Kompatsiaris, Y.; and Li, J., eds., *MM*, 1174–1178.

Casey, M. A.; Veltkamp, R. C.; Goto, M.; Leman, M.; Rhodes, C.; and Slaney, M. 2008. Content-Based Music Information Retrieval: Current Directions and Future Challenges. *Proc. IEEE*, 96(4): 668–696.

Chou, Y.; Chen, I.; Chang, C.; Ching, J.; and Yang, Y. 2021. MidiBERT-Piano: Large-scale Pre-training for Symbolic Music Understanding. *CoRR*, abs/2107.05223.

Dai, Z.; Yang, Z.; Yang, Y.; Carbonell, J.; Le, Q.; and Salakhutdinov, R. 2019. Transformer-XL: Attentive Language Models beyond a Fixed-Length Context. In *ACL*, 2978–2988.

Dettmers, T.; Minervini, P.; Stenetorp, P.; and Riedel, S. 2018. Convolutional 2D Knowledge Graph Embeddings. In McIlraith, S. A.; and Weinberger, K. Q., eds., *AAAI*, 1811–1818.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL*, 4171–4186.

Dong, H.-W.; Hsiao, W.-Y.; and Yang, Y.-H. 2018. Pypianoroll: Open source Python package for handling multitrack pianoroll. *ISMIR*.

Dong, X.; Gabrilovich, E.; Heitz, G.; Horn, W.; Lao, N.; Murphy, K.; Strohmann, T.; Sun, S.; and Zhang, W. 2014. Knowledge vault: a web-scale approach to probabilistic knowledge fusion. In Macskassy, S. A.; Perlich, C.; Leskovec, J.; Wang, W.; and Ghani, R., eds., *SIGKDD*, 601–610.

Ferreira, L. N.; Lelis, L. H. S.; and Whitehead, J. 2020. Computer-Generated Music for Tabletop Role-Playing Games. In Lelis, L.; and Thue, D., eds., *AIIDE*, 59–65.

Foscarin, F.; McLeod, A.; Rigaux, P.; Jacquemard, F.; and Sakai, M. 2020. ASAP: a dataset of aligned scores and performances for piano transcription. In Cumming, J.; Lee, J. H.; McFee, B.; Schedl, M.; Devaney, J.; McKay, C.; Zangerle, E.; and de Reuse, T., eds., *ISMIR*, 534–541.

Gehring, J.; Auli, M.; Grangier, D.; Yarats, D.; and Dauphin, Y. N. 2017. Convolutional Sequence to Sequence Learning. In Precup, D.; and Teh, Y. W., eds., *ICML*, volume 70, 1243–1252.

Hawthorne, C.; Elsen, E.; Song, J.; Roberts, A.; Simon, I.; Raffel, C.; Engel, J. H.; Oore, S.; and Eck, D. 2018. Onsets and Frames: Dual-Objective Piano Transcription. In Gómez, E.; Hu, X.; Humphrey, E.; and Benetos, E., eds., *ISMIR*, 50–57.

He, P.; Liu, X.; Gao, J.; and Chen, W. 2021. Deberta: decoding-Enhanced Bert with Disentangled Attention. In *ICLR*.

Hsiao, W.; Liu, J.; Yeh, Y.; and Yang, Y. 2021. Compound Word Transformer: Learning to Compose Full-Song Music over Dynamic Directed Hypergraphs. In *AAAI*, 178–186.

Huang, C. A.; Vaswani, A.; Uszkoreit, J.; Simon, I.; Hawthorne, C.; Shazeer, N.; Dai, A. M.; Hoffman, M. D.; Dinculescu, M.; and Eck, D. 2019. Music Transformer: Generating Music with Long-Term Structure. In *ICLR*.

Huang, Y.; and Yang, Y. 2020. Pop Music Transformer: Beat-based Modeling and Generation of Expressive Pop Piano Compositions. In Chen, C. W.; Cucchiara, R.; Hua, X.; Qi, G.; Ricci, E.; Zhang, Z.; and Zimmermann, R., eds., *MM*, 1180–1188.

Hung, H.; Ching, J.; Doh, S.; Kim, N.; Nam, J.; and Yang, Y. 2021. EMOPIA: A Multi-Modal Pop Piano Dataset For Emotion Recognition and Emotion-based Music Generation. In Lee, J. H.; Lerch, A.; Duan, Z.; Nam, J.; Rao, P.; van Kranenburg, P.; and Srinivasamurthy, A., eds., *ISMIR*, 318–325.

Jiang, Z.; and Dannenberg, R. B. 2019. Melody identification in standard MIDI files. In *16th Sound & Music Computing Conference*, 65–71.

Ke, G.; He, D.; and Liu, T. 2021. Rethinking Positional Encoding in Language Pre-training. In *ICLR*. OpenReview.net.

Kim, S.; Lee, H.; Park, S.; Lee, J.; and Choi, K. 2020. Deep Composer Classification Using Symbolic Representation. *CoRR*, abs/2010.00823.

Kong, Q.; Choi, K.; and Wang, Y. 2020. Large-Scale MIDI-based Composer Classification. *CoRR*, abs/2010.14805.

Kong, Q.; Li, B.; Song, X.; Wan, Y.; and Wang, Y. 2020. High-resolution Piano Transcription with Pedals by Regressing Onsets and Offsets Times. *CoRR*, abs/2010.01815.

Li, J.; Wang, P.; Li, Z.; Liu, X.; Utiyama, M.; Sumita, E.; Zhao, H.; and Ai, H. 2022. A Fuzzy Training Framework for Controllable Sequence-to-Sequence Generation. *IEEE Access*, 10: 92467–92480.

Lin, Z.; Feng, M.; dos Santos, C. N.; Yu, M.; Xiang, B.; Zhou, B.; and Bengio, Y. 2017. A Structured Self-Attentive Sentence Embedding. In *ICLR*.

Liu, X.; Yu, H.; Dhillon, I. S.; and Hsieh, C. 2020. Learning to Encode Position for Transformer with Continuous Dynamical Model. In *ICML*, volume 119, 6327–6335.

Loshchilov, I.; and Hutter, F. 2019. Decoupled Weight Decay Regularization. In *ICLR*.

Merity, S.; Keskar, N. S.; and Socher, R. 2018. Regularizing and Optimizing LSTM Language Models. In *ICLR*.

Miao, J.; and Yang, Y. 2021. NonupleCLMR: Contrastive Learning of Musical Representation with NonupleMIDI. In Chen, S.; Hu, J.; Li, T.; Martínez, L.; and Liu, J., eds., *ISKE*, 412–419. IEEE.

Nguyen, D. Q.; Nguyen, T. D.; Nguyen, D. Q.; and Phung, D. 2018. A Novel Embedding Model for Knowledge Base Completion Based on Convolutional Neural Network. In *NAACL*, 327–333.

Oore, S.; Simon, I.; Dieleman, S.; Eck, D.; and Simonyan, K. 2020. This time with feeling: learning expressive musical performance. *Neural Comput. Appl.*, 32(4): 955–967.

Qiu, J.; Chen, C. L. P.; and Zhang, T. 2022. A Novel Multi-Task Learning Method for Symbolic Music Emotion Recognition. *CoRR*, abs/2201.05782.

Radford, A.; Narasimhan, K.; Salimans, T.; Sutskever, I.; et al. 2018. Improving language understanding by generative pre-training. *OpenAI Blog*.

Ravishankar, S.; Dewangan, C.; and Talukdar, P. P. 2017. Revisiting Simple Neural Networks for Learning Representations of Knowledge Graphs. In *AKBC@NIPS*.

Shang, C.; Tang, Y.; Huang, J.; Bi, J.; He, X.; and Zhou, B. 2019. End-to-End Structure-Aware Convolutional Networks for Knowledge Base Completion. In *AAAI*, 3060–3067.

Shaw, P.; Uszkoreit, J.; and Vaswani, A. 2018. Self-Attention with Relative Position Representations. In *NAACL*, 464–468.

Sheng, Z.; Song, K.; Tan, X.; Ren, Y.; Ye, W.; Zhang, S.; and Qin, T. 2021. SongMASS: Automatic Song Writing with Pre-training and Alignment Constraint. In *AAAI*, 13798–13805.

Simonetta, F.; Chacón, C. E. C.; Ntalampiras, S.; and Widmer, G. 2019. A Convolutional Approach to Melody Line Identification in Symbolic Scores. In Flexer, A.; Peeters, G.; Urbano, J.; and Volk, A., eds., *ISMIR*, 924–931.

Socher, R.; Chen, D.; Manning, C. D.; and Ng, A. Y. 2013. Reasoning With Neural Tensor Networks for Knowledge Base Completion. In Burges, C. J. C.; Bottou, L.; Ghahramani, Z.; and Weinberger, K. Q., eds., *NeurIPS*, 926–934.

Su, J.; Lu, Y.; Pan, S.; Wen, B.; and Liu, Y. 2021. RoFormer: Enhanced Transformer with Rotary Position Embedding. *CoRR*, abs/2104.09864.

Sun, K.; Li, Z.; and Zhao, H. 2021. Multilingual Pre-training with Universal Dependency Learning. *Advances in Neural Information Processing Systems*, 34: 8444–8456.

Tsai, T.; and Ji, K. 2020. Composer Style Classification of Piano Sheet Music Images Using Language Model Pretraining. In Cumming, J.; Lee, J. H.; McFee, B.; Schedl, M.; Devaney, J.; McKay, C.; Zangerle, E.; and de Reuse, T., eds., *ISMIR*, 176–183.

Vashishth, S.; Sanyal, S.; Nitin, V.; Agrawal, N.; and Talukdar, P. P. 2020. InteractE: Improving Convolution-Based Knowledge Graph Embeddings by Increasing Feature Interactions. In *AAAI*, 3009–3016.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is All you Need. In Guyon, I.; von Luxburg, U.; Bengio, S.; Wallach, H. M.; Fergus, R.; Vishwanathan, S. V. N.; and Garnett, R., eds., *NeurIPS*, 5998–6008.

Verma, H.; and Thickstun, J. 2019. Convolutional Composer Classification. In Flexer, A.; Peeters, G.; Urbano, J.; and Volk, A., eds., *ISMIR*, 549–556.

Wang, B.; Zhao, D.; Lioma, C.; Li, Q.; Zhang, P.; and Simonsen, J. G. 2020a. Encoding word order in complex embeddings. In *ICLR*.

Wang, Z.; Chen, K.; Jiang, J.; Zhang, Y.; Xu, M.; Dai, S.; and Xia, G. 2020b. POP909: A Pop-Song Dataset for Music Arrangement Generation. In Cumming, J.; Lee, J. H.; McFee, B.; Schedl, M.; Devaney, J.; McKay, C.; Zangerle, E.; and de Reuse, T., eds., *ISMIR*, 38–45.

Wang, Z.; and Xia, G. 2021. MuseBERT: Pre-training Music Representation for Music Understanding and Controllable Generation. In Lee, J. H.; Lerch, A.; Duan, Z.; Nam, J.; Rao, P.; van Kranenburg, P.; and Srinivasamurthy, A., eds., *ISMIR*, 722–729.

Yang, B.; Yih, W.-t.; He, X.; Gao, J.; and Deng, L. 2014. Embedding entities and relations for learning and inference in knowledge bases. *arXiv preprint arXiv:1412.6575*.

Zeng, M.; Tan, X.; Wang, R.; Ju, Z.; Qin, T.; and Liu, T.-Y. 2021. MusicBERT: Symbolic Music Understanding with Large-Scale Pre-Training. In *Findings of ACL-IJCNLP*, 791–800.