SYMPHONY GENERATION WITH PERMUTATION INVARIANT LANGUAGE MODEL

Jiafeng Liu[1⋆] **Yuanliang Dong**[1⋆] **Zehua Cheng**[2]
Xinran Zhang[1] **Xiaobing Li**[1] **Feng Yu**[1] **Maosong Sun**[1,3†]
[1] Department of Music AI and Music Information Technology, Central Conservatory of Music
[2] Department of Computer Science, University of Oxford
[3] Department of Computer Science and Technology , Tsinghua University
{jiafeng.liu, gunterdong}@mail.ccom.edu.cn

ABSTRACT

In this work, we propose a permutation invariant language model, SymphonyNet, as a solution for symbolic symphony music generation. We propose a novel Multi-track Multi-instrument Repeatable (MMR) representation for symphonic music and model the music sequence using a Transformer-based auto-regressive language model with specific 3-D positional embedding. To overcome length overflow when modeling extra-long symphony tokens, we also propose a modified Byte Pair Encoding algorithm (Music BPE) for music tokens and introduce a novel linear transformer decoder architecture as a backbone. Meanwhile, we train the decoder to learn automatic orchestration as a joint task by masking instrument information from the input. We also introduce a large-scale symbolic symphony dataset for the advance of symphony generation research. Empirical results show that the proposed approach can generate coherent, novel, complex and harmonious symphony as a pioneer solution for multi-track multi-instrument symbolic music generation.

1. INTRODUCTION

Symphony is one of the most complex and brilliant musical composition forms in human history, where many instruments are intertwined to express rich human emotions. The past decade has seen the rapid development and tremendous success of the symbolic music generation in both research and industrial field [1–3]. Most current works follow conventional text modeling and generation method by applying language model to sequences of symbolic musical events [4–6]. However, symphony modeling and generation still constitutes in itself a considerable challenge since symphony music sequences differ from text sequences in various aspects.

Natural language could be modeled as a purely linear sequence constructed strictly by a sequential order of words. Symphony scores, on the other hand, are usually viewed as two-dimensional symbolic sequences in which many notes can be played concurrently. Notes in a symphony score are **semi-permutation invariant**. More specifically, as shown in Fig. 1, the blue box indicates the musical instrument tracks, and the corresponding staves on the right side are permutation invariant. Similarly, the notes inside the red box are also permutation invariant. In contrast, notes in the upper yellow box are permutation variant since each note is played sequentially. Changes in the order of notes will impair the music itself. The yellow box at the bottom is a more complicated situation: a permutation variant note sequence in general containing permutation invariant notes. Simply flattening the score into a 1-D text-like sequence may damage the local structure of music [7]. To address this problem, we propose the Multi-track Multi-instrument Repeatable (MMR) representation with particular 3-D positional embedding in Section 3 which fully considers the properties of semi-permutation invariance in symbolic music scores.

Moreover, when comparing music scores with text, conventionally notes could be considered as characters, while intervals or chords are comparable to words. Modeling musical events at note level is a common practice [5,6,8,9]. However, this may be confronted with similar problems in char-level text generation, such as extremely long sequences and less meaningful individual tokens. Word-level tokenization suffers from large vocabulary size and out of vocabulary (OOV) problems. Byte Pair Encoding (BPE) [10, 11] subword tokenization is a tradeoff between word-level and character-level tokenization. Inspired by BPE, we propose the Music BPE algorithm in Section 4, which could automatically aggregate notes to intervals and chords as subwords without a pre-defined vocabulary and construct music sequences with richer semantics.

Generating symphony music with proper instruments for different tracks is another challenging task. Recent

⋆ The authors contributed equally to this work.
† Corresponding author. Email: sms@tsinghua.edu.cn

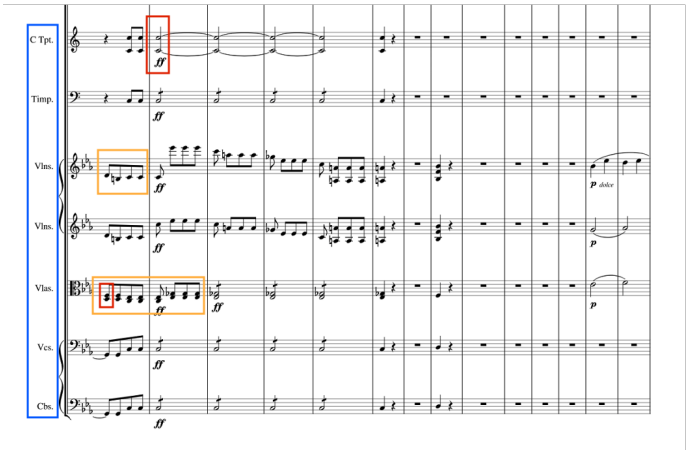

Figure 1: A simple example of Multi Instruments & Multi Tracks & Repeat Instruments symphony score.

work like Arranger [12] focuses on instrumentation by learning to separate parts from the mixture in symbolic multi-track music. However, it does not incorporate music generation task. In this paper, we present a unique linear transformer decoder architecture for instrument classification with joint-task training, which allows the model to learn auto-orchestration rather than relying on instrument information as an pre-defined input source. [5, 13–15].

The contributions of this paper are presented as below:

- We propose a novel Multi-track Multi-instrument Repeatable (MMR) representation for symphony music, including particular 3-D positional embedding designed to address the semi-permutation invariant challenge in symphony generation. Our method is also compatible with all existing symbolic music ensembles, including but not limited to piano solo, quartet and pop band music.
- We propose a novel algorithm, Music BPE, to model the symbolic music at subword-level. Furthermore, we found that our Music BPE algorithm could aggregate notes to intervals and chords, which are consistent with common chords summarized by human musicians.
- We introduce SymphonyNet, a novel music generation model with joint-task training for instrument classification based on our proposed MMR representation and Music BPE. The model can learn the proper orchestration according to the distribution of the notes.
- We collect a symphony MIDI dataset, consisting of $46,359$ high-quality MIDI files with multiple instruments and tracks to advance researches on symphony generation with deep learning.

2. RELATED WORK

We organize some existing works in Table 1 in terms of five aspects of symbolic music modeling: time unit, representation method, backbone model, music type and the ability to model music with repeat instruments. Generation works are presented above and understanding works are presented below. Pianoroll, MIDI event timeshift, and Beat-based onset and duration are the mainstream time units in music generation and understanding tasks. However, Pianoroll divides music into fixed-length grids, and MIDI format provides overprecise timeshift events, both suffering from sparsity problems, which raises another handicap for applying deep learning models in this multi-track generation. Pop Music Transformer [8] is the first attempt to introduced the beat-based REMI representation in music generation. It supports variable-length duration of notes, which is more musically inspired. Compound Word [6], derived from REMI representation, classifies the sequence of REMI into note-related or metric-related events, which are then aggregated, greatly decreasing the sequence length.. This has engendered a new trend of beat-based symbolic music generation.

Language models are now prevalent in natural language processing tasks [18]. However, applying language models to the creation of multi-track music remains challenging. MuMIDI [5] and OctupleMIDI [9] models multiple attributes of one note in one sequence step and also incorporates instrument tokens for multi-track representation. However, if one musical piece contains more than one track for the same instrument, their representation could not distinguish them in different tracks. MMM [15] introduced a MIDI-event-like representation, creating a time-ordered sequence of musical events for each track and concatenating several tracks into a single sequence. However, MMM adopts time-delta tokens and fixed positional encoding which weakens the note-level correlation and structure between tracks. MuseBert [7] proposes a permutation invariant bert-like language model with generalized relative position encoding (RPE) which, however, is not compatible with multi-track music generation.

Though various symbolic music representation strategies have been proposed, few are compatible with multi-track music with repeatable instruments or tracks, such as the symphony. Furthermore, permutation invariance of music, as is discussed in Section 1, has scarcely been considered. To our knowledge, this work proposes the first representation and tokenization method to encode music with multiple repeatable instruments and multiple repeatable tracks and designs a universal and effective strategy for generating symphony music with permutation invariant language model.

3. MULTI-TRACK MULTI-INSTRUMENT REPEATABLE REPRESENTATION

To further analyze the symphony generation task, it is crucial to understand the difference between the symphony format and other genres of music.

- **Single Instrument in Single Track**. No more than one note is played at any timestep by one instrument. Also called monophonic music. e.g., flute.
- **Multi Instruments & Each in Single Track**. Only one

Name	Time Unit	Representation	Backbone Model	Type of Music	Instruments
DeepBach [1]	Fixed-length grid	N/A	Bi-directional RNN	Chorales	Fixed Ensemble
MuseGAN [16]	Fixed-length grid	N/A	GAN	Multi-track	Fixed Ensemble
Music Transfor. [4]	MIDI-event timeshift	N/A	Vanilla Transformer	Piano	N/A
Pop MT [8]	Beat and note duration	REMI	Transformer-XL	Piano	N/A
CWT [6]	Beat and note duration	Compound Word	Linear Transformer	Piano	N/A
Musenet [13]	MIDI-event timeshift	Token Aggregation	GPT-2	Multi-track	Not Repeatable
PopMAG [5]	Beat and note duration	MuMIDI	Transformer-XL	Multi-track	Not Repeatable
LakhNES [14]	MIDI-event timeshift	Token Aggregation	Transformer-XL	Multi-track	Fixed Ensemble
MMM [15]	MIDI-event timeshift	Hierarchical	GPT-2	Multi-track	Repeatable
This work	Beat and note duration	MMR	Linear Transformer	Multi-track	Repeatable
PiRhDy [17]	Fixed-length grid	Fusion module	RNN with attention	Multi-track	Not Repeatable
MusicBert [9]	Beat and note duration	OctupleMIDI	Roberta	Multi-track	Not Repeatable

Table 1: An overview of time unit, representation, backbone model and music type in existing works, above for generation works and below for understanding works.

note for each instrument is played at any timestep. e.g., quartet singing.

- **Single Instrument in Multi Tracks**. There are multiple notes played in each timestep while only one instrument. e.g., piano.
- **Multi Instruments & Multi Tracks & No Repeat Instrument**. There are multiple notes played in each timestep. No constraint on the number of instruments and all instruments are unique. e.g., classical pop band with only drum, electric guitar and bass.
- **Multi Instruments & Multi Tracks & Repeat Instruments**. Instruments are not unique and multiple same instruments can play different notes on different tracks, e.g., symphony.

For the last case, it's a common practice to merge the same instruments into a single track in previous works. However, it may damage the intrinsic structure of symphony music. For example, this may cause a violin to play polyphonic notes, or even intermingle multiple melody lines. Our proposed Multi-track Multi-instrument Repeatable (MMR) representation models repeated instruments separately, which could capture more heuristic musical information within a single track. Since our MMR representation is aimed at symphony modeling, it is also compatible with all existing music ensembles.

3.1 Structural and Controlling Token

We consider that special tokens perform two primary functions in a symphony music generation task: 1) To represent the musical structures of notes. 2) To control the model output during the inference phase.

Score We use a pair of $[BOS]$ and $[EOS]$ tokens to designate the beginning and end of a symphony score.

Measure Different from [5, 8], we ascribe a pair of $[BOM_i]$ and $[EOM]$ to indicate the beginning and end of a measure, the character i to represent the total length of the current measure. The length of a measure is calculated by time signature, and we choose 32^{th} note as the smallest unit of time. For example, a 4/4 time signature indicates four quarter notes length per measure, which is equal to the length of thirty-two 32^{th} notes. In that case, character i equals 32, and the measure beginning token is $[BOM_{32}]$

Chord The chord token is a valuable indicator of how generally notes are arranged in the current measure. We pre-define 132 common types of chord token and pre-compute chord tokens with a rule-based algorithm proposed in [6], such as C major seventh chord marked as token $[C_{maj7}]$.

Track Unlike any previous works, we do not explicitly encode the track and instrument transformation in a single token. A change track token $[CC]$ only signifies the start of a new track for the latter controlling purpose. Section 5 will further explore the traits of tracks and instruments and the approaches of differentiating tracks.

Position A position token stands for the *onset* of a note within the measure, represented by the token $[POS_j]$. The following event tokens are controlled by the current position token until another position token shows up. The character j means the number of the current unit of time position. For example, a $[POS_{48}]$ indicates the 48^{th} unit time position.

To summarize, structural and controlling tokens are designed to specify the general time-spatial features of notes, such as the time a note is to be played and the track it locates. In this work, these tokens are mandated with a sequential order, as a *measure* token shall be followed by a *chord* token, which altogether represents in a explicit way the measure order as shown in Fig. 3.

3.2 Note-Related Tokens

A note in music scores could be defined in five attributes: pitch, duration, onset, track and instrument. Pitch and duration are content-related and the others are position-related. The latter will be discussed in Section 5. To avoid the long-tail problem, we regard pitch and duration to be distinct note properties and construct two separate vocabularies for model input. Then we aggregate note pitches with identical duration and onset by our proposed Music

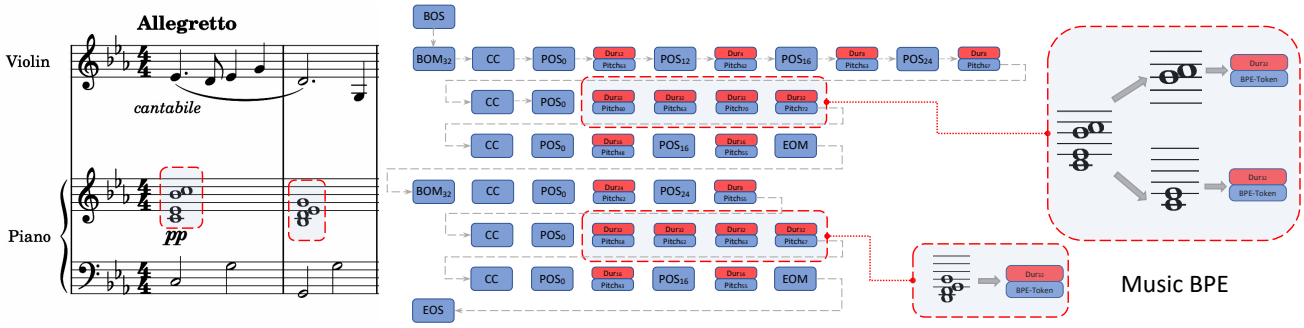

Figure 2: An example of MMR representation and illustration of Music BPE process

BPE algorithm, as will be described in the next section.

4. MUSIC BYTE PAIR ENCODING

As shown in Fig. 2, a complex chord is constructed by several notes at the same position in a measure, which can be deconstructed into two common and simple intervals. Unlike natural language, notes played at the same position are permutation invariant. Changing the order of notes in a chord does not affect its sound or meaning. For instance, a Chord C consists of ($C4$, $E4$, $G4$), which is equal to ($G4$, $C4$, $E4$). This intrinsic property may conflict with the typical natural language processing job, imposing a new constraint on the use of conventional tokenization methods such as standard BPE.

In this work, we propose a novel encoding approach, Music Byte Pair Encoding (Music BPE), for multi-track symbolic music sequence tokenization and preprocessing to exploit the semantics of music events and minimise the length of the input context from a representation standpoint. Different from the original BPE algorithm, our proposed Music BPE is based on **concurrence** of notes rather than **adjacency** of characters.

Our implementation of Music BPE is shown in Algorithm 1. As is mentioned in Section 1, a note has five attributes: *pi*tch, *du*ration, *po*sition, *tr*ack and instrument, while the instrument depends utterly on track within the same measure. Formally, in a piece of symbolic music, we define a maximum set of two or more notes

$$\{(pi, du, po, tr) \mid \text{where } du, po, tr \text{ is constant}\}$$

as a *mulpi* (multiple pitches), i.e., a maximum set of notes that have the same duration at the same global position and within the same track, equivalent to a "word" in the BPE algorithm.

We collect notes with the same global position and the same duration in the same track from each music piece to construct a bag of *mulpies*. The vocabulary list is initialized with 128 MIDI pitches, where each token represents a pitch-set containing a single pitch. Every time we locate all concurrent pairs of tokens in the bag of *mulpies*, merge the most frequent pair ('P_1', 'P_2') into a new symbol P and replace the pair with the new symbol in each *mulpi* until the vocabulary size reaches the maximum limit. A further discussion on the results of the Music BPE algorithm and its effectiveness on our symphony dataset will be presented in Section 5.

Algorithm 1 Music BPE

Input: A multi-set of *mulpies* B
Parameter: desired dictionary size n
Output: Merged dictionary V

Let $V = \{\{p\} \mid p \in [0, 128)\}$.
Let C be an empty multi-set
for all $mulpi \in B$ **do**
  $mulpi \leftarrow \{\{p\} \mid p \in mulpi\}$
  for all $\{P_1, P_2\} \subseteq mulpi$ **do**
    Insert (P_1, P_2) into C.
  end for
end for
while $|V| < n$ **do**
  Let (P_1, P_2) be the most frequent pair in C.
  $V \leftarrow V \cup \{P_1 \cup P_2\}$
  for all $mulpi \in B$ **do**
    if $\{P_1, P_2\} \subseteq mulpi$ **then**
      for all $Q \in mulpi - \{P_1, P_2\}$ **do**
        Delete $(Q, P_1), (Q, P_2)$ from C.
        Insert $(Q, P_1 \cup P_2)$ into C.
      end for
      $mulpi \leftarrow (mulpi - \{P_1, P_2\}) \cup \{P_1 \cup P_2\}$
    end if
  end for
end while
return V

5. SYMPHONYNET DETAILS

5.1 The 3-D Positional Embedding

Transformer [19] is the most used backbone for language model, which is designed *permutation invariant*: if the positional encoding is not added, disrupting the order of the inputs will yield the same output, for transformer model treats inputs as a *set* during self-attention. Therefore, considering this property of Transformer, we design a particular 3-D positional embedding to represent such a semi-permutation invariant feature as shown in Fig. 3. Event tokens follow a semi-permutation variant order on both the measure order and the note order axes. For example, notes played on the same position share the same note positional

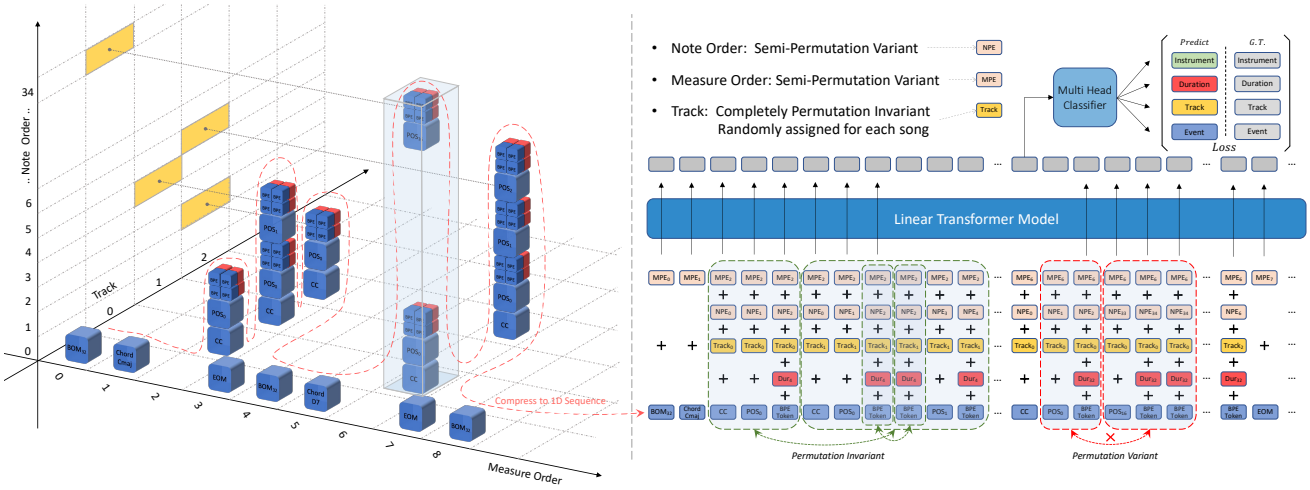

Figure 3: A illustration of the spatial and structural attributes of MMR sequence (left) and the way it is compressed and organized as model input (right).

embedding. In contrast, the track axis is entirely permutation invariant since we only need track embeddings to differentiate tracks other than a sequential order. We use the red curves to illustrate the musical moving trajectory of event tokens to better understand how we compress the spatial and structural sequence of the event tokens into one dimension and send them to the model. At last, we add all constructed embeddings vertically as the model input. To address the extraordinarily long symbolic music sequences challenge, we employ the linear transformer [20] as the backbone of our model to satisfy the length constraint. The model follows a decoder-only fashion, and we design different feed-forward heads for four attributes of musical events, which are **Instrument, Track, Duration, and Event** tokens as shown in Fig. 3.

5.2 Joint Task with Instrument Classification

We mask instrument information for every input token at the input side, and anticipate that the model will learn instrumentation from the output side with instrument loss. This will turn a succession of simple, blank notes into a fully orchestrated piece of music, analogous to colouring a black-and-white painting. This design is motivated by two primary concerns. First, we investigate the possibility if other instrument may play a certain instrument's note track. Therefore, that is a case for the model to determine to what degree the instrument fits the track's notes and how instruments interact with one another across tracks. For instance, it is allowed to substitute the piano for the marimba in some musical compositions. The intrinsic nature of a pre-assigned instrument for notes reduces the diversity of training data.

6. EXPERIMENTS AND RESULTS

This section introduces the novel symphony dataset we propose and presents two stages in the training process [1] . Secondly, we describe controllable methods to generate music under certain condition before we provide findings from Music BPE and compare them with the specific musical knowledge. Lastly, a human evaluation result analysis and scoring on several excerpts generated by different models will be presented.

6.1 Symphony Dataset

To tackle the obstacles of the symphony generation research, we gather a big corpus of symphonic music from multiple online sites and conduct a extensive data cleaning. The average duration of the 46,359 MIDI files containing multiple instruments and tracks, mostly symphony, is 4.26 minutes. The collection contains more than 279 million notes and 567 million tokens in MMR forms. Our symphony dataset is, to the best of our knowledge, the first worldwide large-scale symbolic symphonic music dataset, which might serve as a foundation for future work in multi-track music production.

6.2 Training Details

In our experiment, the model adopts 4096 as the length of input sequence. We set the embedding size for event tokens, durations, instruments and 3-D positional embedding to 512. The final size of event token vocabulary is assigned to 1000 after running Music BPE algorithm and the vocabulary size of durations, instruments, 3-D positional embeddings are derived from the dataset. The linear transformer decoder contains 12 self-attention layers and each layer consists of 16 attention heads. SymphonyNet is trained with eight 2080 Ti GPU and we use a batch size of 128 and an AdamW [21] optimizer with a learning rate of 3×10^{-4}.

6.3 Music BPE Result

After constructing a vocabulary list of length $1,000$ with Music BPE algorithm, the top-5 merged pairs shown in Fig. 4 with the highest frequency are: $(D4, F4)$, $(C4, E4)$,

[1] Our code, demos, dataset and further analysis can be accessed at https://symphonynet.github.io

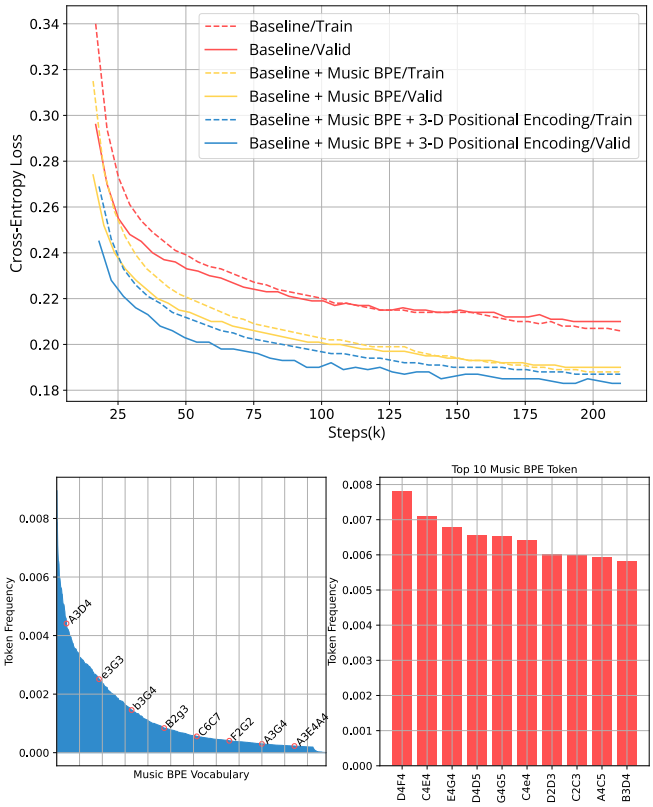

Figure 4: The training and validation curves of different models and the Music BPE note aggregation results.

	Model	C	D	H	S	O	P
Chord	Baseline	3.5	3.57	3.07	3.00	3.21	3.29
	BPE	3.64	3.64	3.14	**3.15**	3.43	3.29
	3D + BPE	**3.71**	**3.72**	**3.21**	3.07	**3.5**	**3.5**
	Human	4.43	3.43	4.14	4.36	4.14	4.14
Prime	Baseline	3.79	2.79	3.21	3.43	3.36	3.36
	BPE	**3.86**	**3.5**	**3.5**	3.5	3.64	**3.86**
	3D + BPE	**3.86**	3.14	3.43	**3.57**	**3.93**	3.64
	Human	4.36	3.57	4.36	4.00	4.36	4.36
Uncondi.	Baseline	3.52	3.46	3.04	3.07	3.11	3.07
	BPE	**3.79**	3.64	3.25	3.11	3.25	3.29
	3D + BPE	3.53	**3.93**	**3.43**	**3.32**	**3.43**	**3.32**
	Human	4.39	3.89	4.18	4.21	4.11	4.29

(a) Trained on Symphony Dataset

Model	C	D	H	S	O	P
MMM	3.20 ±0.13	2.71 ±0.12	2.51 ±0.13	2.66 ±0.12	2.80 ±0.13	2.71 ±0.11
Symph.	**3.33** ±0.15	**2.89** ±0.13	**2.76** ±0.12	**2.69** ±0.13	**2.99** ±0.12	**2.87** ±0.13

(b) Trained on Lakh MIDI Dataset

Table 2: Human evaluation results from 25 musicians and 25 non-musicians, with mean opinion scores and 95 percent confidence intervals reported.

($E4$, $G4$), ($D4$, $D5$), and ($G4$, $G5$), which are usual intervals occurring in symphony *divisi* passages. After applying Music BPE on the whole music corpus, the average token length of a $mulpi$ shortens to half (from 2.28 to 1.13), also reducing the burden of modeling ultra-long symphony sequences.

6.4 Ablation Study and Human Evaluations

We train a linear transformer decoder model with the vanilla positional encoding of GPT-3 [18] as a baseline. Then we train another model of the same architecture with the training data processed by the proposed Music BPE algorithm. Finally, we incorporate both 3-D positional embedding and Music BPE algorithm, which achieves the lowest training and validation loss after the same total training steps, as is shown in Fig. 4. The objective metric indicates that our permutation-invariant 3-D positional embedding and Music BPE algorithm could significantly improve model performance and generalization ability.

Also, we perform a human evaluation to compare the quality of generated music excerpts from different models with human composition. Participants include 25 professional musicians and 25 non-musicians. Each participant receives 16 excerpts: four excerpts conditioned on a chord progression, four excerpts conditioned on a given 4-bar prime, and eight unconditioned excerpts. The music excerpts are rated in 5 dimensions: Coherence **(C)**, Diversity **(D)**, Harmoniousness **(H)**, Structureness **(S)**, Orchestration **(O)** and Overall Preference **(P)**, in a 5-point Likert scale. As shown in Table 2a, the model with 3-D positional embedding and Music BPE beats most of the approaches. It is worth noting that excerpts generated by our models surpass the human compositions in the indicator of diversity marked by yellow color.

To further explore the model performance, we retrain SympohyNet on Lakh MIDI Dataset [22] with the same backbone model architecture as MMM [15], and carry out another human evaluation to compare with MMM. Each participant receives 10 excerpts: five generated unconditionally from MMM and the others generated unconditionally from retrained SymphonyNet. The results are presented in Table 2b, which indicate that SymphonyNet surpass MMM in all indicators. Overall, the human hearing test suggests that SymphonyNet can construct coherent, unique, complex, and harmonic symphonies.

7. CONCLUSION

In this work, we illustrate the properties of multi-track and multi-instrument music, like symphony, and propose a novel MMR representation with 3-D positional embedding for modelling it. To tokenize the ultra-long symbolic music sequence at sub-word level, we propose the Music BPE algorithm. Besides, we design a joint task for the model to learn auto-orchestration. Human evaluation results show that our suggested technique produces competitive symphonic music when compared to human compositions. In the future, we will investigate modelling long-term musical structures, since complex music, such as symphonies, often consists of numerous parts or movements.

8. ACKNOWLEDGEMENTS

Thanks for the anonymous reviewers for their valuable comments. This work is supported by High-grade, Precision and Advanced Discipline Construction Project of Beijing Universities, Major Projects of National Social Science Fund of China (Grant No. 21ZD19), and Nation Culture and Tourism Technological Innovation Engineering Project.

9. REFERENCES

[1] G. Hadjeres, F. Pachet, and F. Nielsen, "DeepBach: a steerable model for bach chorales generation," in *International Conference on Machine Learning*. PMLR, 2017, pp. 1362–1371.

[2] L. Yang, S. Chou, and Y. Yang, "MidiNet: A convolutional generative adversarial network for symbolic-domain music generation," in *Proceedings of the 18th International Society for Music Information Retrieval Conference, ISMIR 2017, Suzhou, China, October 23-27, 2017*, 2017, pp. 324–331.

[3] A. Roberts, J. Engel, C. Raffel, C. Hawthorne, and D. Eck, "A hierarchical latent vector model for learning long-term structure in music," in *International conference on machine learning*. PMLR, 2018, pp. 4364–4373.

[4] C.-Z. A. Huang, A. Vaswani, J. Uszkoreit, I. Simon, C. Hawthorne, N. Shazeer, A. M. Dai, M. D. Hoffman, M. Dinculescu, and D. Eck, "Music transformer: Generating music with long-term structure," in *International Conference on Learning Representations*, 2018.

[5] Y. Ren, J. He, X. Tan, T. Qin, Z. Zhao, and T.-Y. Liu, "Popmag: Pop music accompaniment generation," in *Proceedings of the 28th ACM International Conference on Multimedia*, 2020, pp. 1198–1206.

[6] W.-Y. Hsiao, J.-Y. Liu, Y.-C. Yeh, and Y.-H. Yang, "Compound word transformer: Learning to compose full-song music over dynamic directed hypergraphs," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021, pp. 178–186.

[7] Z. Wang and G. Xia, "MuseBERT: Pre-training music representation for music understanding and controllable generation," in *Proceedings of the 22nd International Society for Music Information Retrieval Conference, ISMIR 2021, Online, November 7-12, 2021*, 2021.

[8] Y.-S. Huang and Y.-H. Yang, "Pop music transformer: Beat-based modeling and generation of expressive pop piano compositions," in *Proceedings of the 28th ACM International Conference on Multimedia*, 2020, pp. 1180–1188.

[9] M. Zeng, X. Tan, R. Wang, Z. Ju, T. Qin, and T. Liu, "Musicbert: Symbolic music understanding with large-scale pre-training," in *Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021, Online Event, August 1-6, 2021*, ser. Findings of ACL, vol. ACL/IJCNLP 2021. Association for Computational Linguistics, 2021, pp. 791–800. [Online]. Available: https://doi.org/10.18653/v1/2021.findings-acl.70

[10] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2016, pp. 1715–1725.

[11] P. Gage, "A new algorithm for data compression," *The C Users Journal*, vol. 12, no. 2, pp. 23–38, 1994.

[12] H. Dong, C. Donahue, T. Berg-Kirkpatrick, and J. J. McAuley, "Towards automatic instrumentation by learning to separate parts in symbolic multitrack music," in *Proceedings of the 22nd International Society for Music Information Retrieval Conference, ISMIR 2021, Online, November 7-12, 2021*, 2021.

[13] C. Payne, "MuseNet," *OpenAI Blog*, vol. 3, 2019.

[14] C. Donahue, H. H. Mao, Y. E. Li, G. W. Cottrell, and J. J. McAuley, "Lakhnes: Improving multi-instrumental music generation with cross-domain pre-training," in *Proceedings of the 20th International Society for Music Information Retrieval Conference, ISMIR 2019, Delft, The Netherlands, November 4-8, 2019*, 2019.

[15] J. Ens and P. Pasquier, "Mmm: Exploring conditional multi-track music generation with the transformer," *arXiv preprint arXiv:2008.06048*, 2020.

[16] H.-W. Dong, W.-Y. Hsiao, L.-C. Yang, and Y.-H. Yang, "Musegan: Multi-track sequential generative adversarial networks for symbolic music generation and accompaniment," in *Thirty-second aaai conference on artificial intelligence*, 2018.

[17] H. Liang, W. Lei, P. Y. Chan, Z. Yang, M. Sun, and T.-S. Chua, "Pirhdy: Learning pitch-, rhythm-, and dynamics-aware embeddings for symbolic music," in *Proceedings of the 28th ACM International Conference on Multimedia*, 2020, pp. 574–582.

[18] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language models are few-shot learners," *arXiv preprint arXiv:2005.14165*, 2020.

[19] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[20] A. Katharopoulos, A. Vyas, N. Pappas, and F. Fleuret, "Transformers are rnns: Fast autoregressive transformers with linear attention," in *International Conference on Machine Learning*. PMLR, 2020, pp. 5156–5165.

[21] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *International Conference on Learning Representations*, 2018.

[22] C. Raffel, "Learning-based methods for comparing sequences, with applications to audio-to-midi alignment and matching," Ph.D. dissertation, COLUMBIA UNIVERSITY, 2016.