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$^{\text{th}}$ 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$^{\text{th}}$ 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 precompute 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$^{\text{th}}$ 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

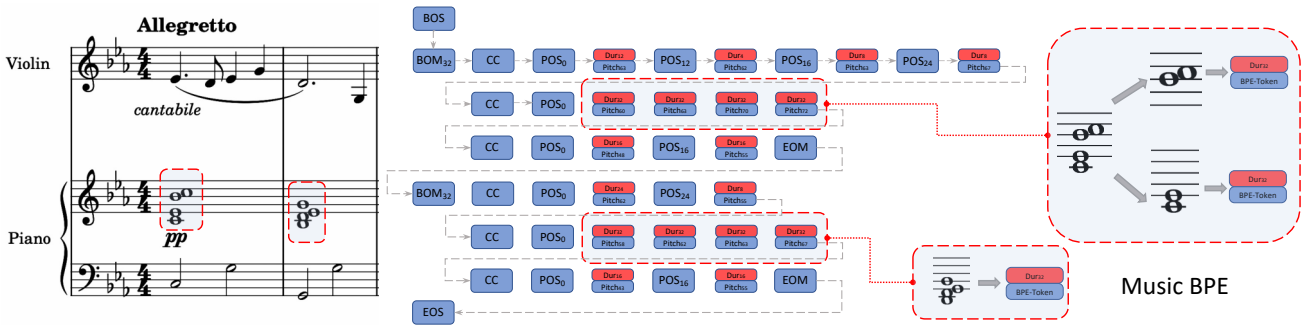

Figure 2: An example of MMR representation and illustration of Music BPE process

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

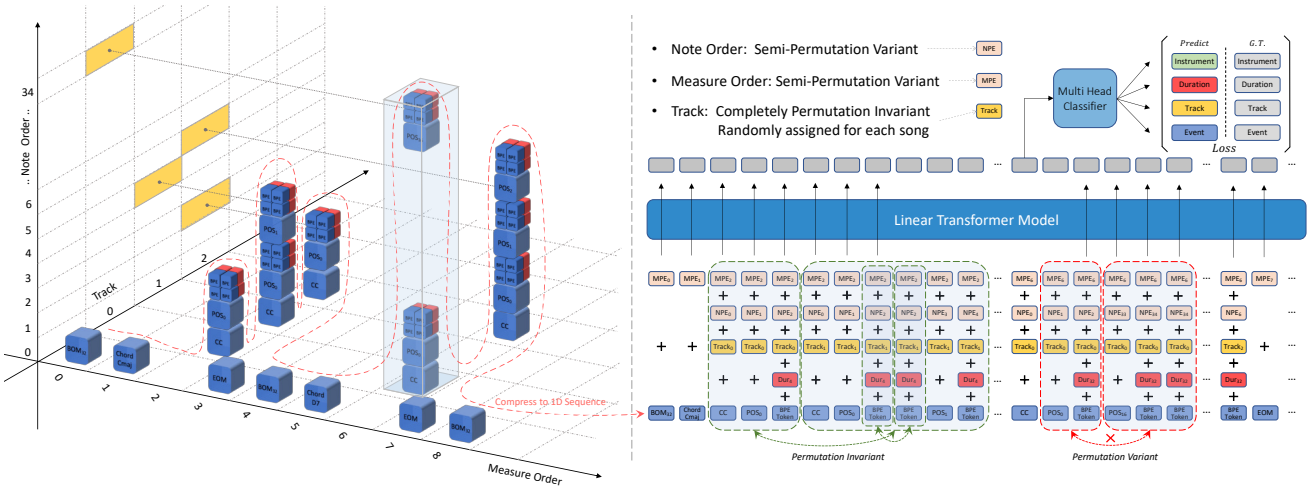

Figure 3: A illustration of the spatial and structural attributes of MMR sequence (left) and the way it is compressed and organized as model input (right).

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

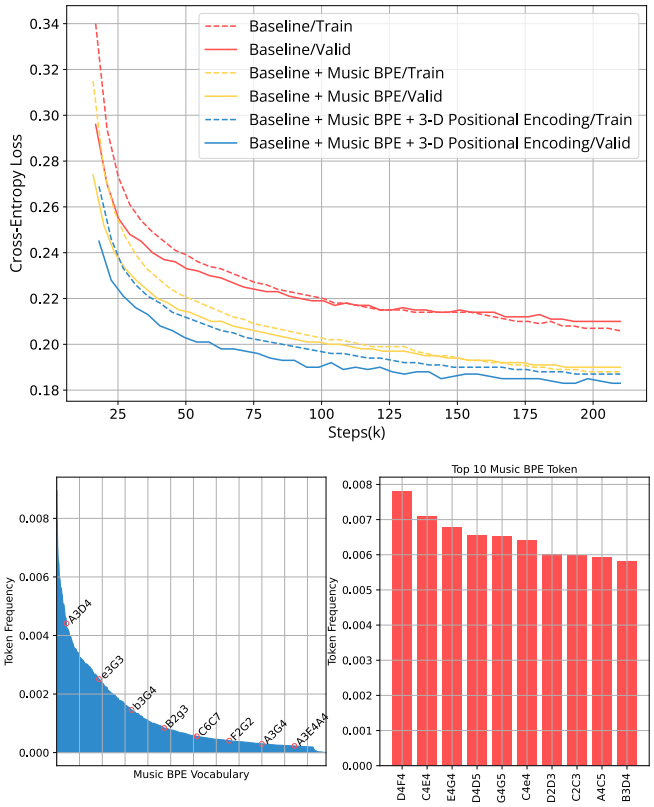

Figure 4: The training and validation curves of different models and the Music BPE note aggregation results.

	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.

MULAN: A JOINT EMBEDDING OF MUSIC AUDIO AND NATURAL LANGUAGE

Qingqing Huang[1] **Aren Jansen**[1] **Joonseok Lee**[1,2]
Ravi Ganti[1] **Judith Yue Li**[1] **Daniel P. W. Ellis**[1]
Google Research[1], Seoul National University[2]
{qqhuang, arenjansen, joonseok, gmravi, judithyueli, dpwe}@google.com

ABSTRACT

Music tagging and content-based retrieval systems have traditionally been constructed using pre-defined ontologies covering a rigid set of music attributes or text queries. This paper presents MuLan: a first attempt at a new generation of acoustic models that link music audio directly to unconstrained natural language music descriptions. MuLan takes the form of a two-tower, joint audio-text embedding model trained using 44 million music recordings (370K hours) and weakly-associated, free-form text annotations. Through its compatibility with a wide range of music genres and text styles (including conventional music tags), the resulting audio-text representation subsumes existing ontologies while graduating to true zero-shot functionalities. We demonstrate the versatility of the MuLan embeddings with a range of experiments including transfer learning, zero-shot music tagging, language understanding in the music domain, and cross-modal retrieval applications.

1. INTRODUCTION

Classifiers are generally trained to label examples with predefined and fixed class inventories, which are often manually specified as a structured ontology indicating interclass relationships. Empowered by recent advances in neural language modeling and their demonstrated transfer learning competence, researchers have begun exploring less restrictive natural language interfaces to access the categorical information underlying raw content signals. The majority of this work has been in the visual and audio event domain, where a recent series of studies have demonstrated the utility of jointly embedding media content with natural language captions [1–5]. These joint embeddings have demonstrated strong capabilities in a range of applications, including transfer learning, cross-modal retrieval, automatic captioning, and zero-shot classification.

The success of these efforts strongly depends on large-scale training resources and hefty neural network architectures that are flexible enough to model the complex, non-monotonic relationship between language and other modalities. In particular, the visual domain has greatly benefited from the availability of large amounts of captioned images available across the web [1]. However, in the general environmental audio domain, such large-scale audio-caption pairs are less readily available and related efforts have relied on small captioned datasets [6, 7]. Critically, these datasets do not span the diversity of sound-descriptive language and their success in the more difficult zero-shot setting has been lacking [3, 8, 9].

This paper considers this task of jointly embedding audio and natural language, but focuses specifically on the music domain. Our goal is to produce a flexible language interface with which any musical concept can be linked to related music audio. We face similar training data prerequisites to works listed above. However, while general environmental audio consists of background sounds that are unlikely to elicit unprompted description, music audio is often a central focus. Consequently, text associated with music videos is much more likely to relate to the underlying musical concepts that we aim to model (e.g., genres, artists, moods, structure). Thus, our strategy is to assemble a collection of textual annotations extracted from metadata, comments, and playlist data and map them to a training set of over 44 million internet music videos. As was the case with image-text model training in [1], our text data only truly refers to the musical content in a fraction of cases. Therefore, we also explore text pre-filtering using a text classifier separately trained to identify music descriptions.

We use this large-scale dataset to train MuLan, a new generation of semantically-structured music audio embedding model equipped with a natural language interface. MuLan employs a two-tower parallel encoder architecture, using a contrastive loss objective that elicits a shared embedding space between music audio and text. We demonstrate that MuLan not only leads to state-of-the-art performance in transfer learning for various music information retrieval tasks, but also enables a range of functionalities in cross-modal text-to-music retrieval, zero-shot music tagging, and music-domain language understanding.

2. RELATED WORK

Audio representation learning. Transfer learning using large-scale, task-agnostic pretraining of general-purpose content representations has become a dominant approach

in several fields. Audio representation learning has been no exception, including both general environmental audio [10, 11] and music audio [12–15]. Different pretraining mechanisms have been explored. In supervised pretraining, an Audio Spectrogram Transformer (AST) [10], pretrained on ImageNet [16] and AudioSet [17], achieved state-of-the-art results in various tagging tasks. A strong early baseline for music audio representation learning was provided in [13], using the Million Song Database [18].

In unsupervised and self-supervised pretraining, both discriminative and generative model approaches have been demonstrated to be successful. Discriminative training was explored in [19–22] where the models tried to learn representations that assign higher similarity to audio segments extracted from the same recording compared to segments from different recordings. SSAST [11] explored similar discriminative losses, as well as generative masked spectrogram patch modeling. It was shown in [23] that the intermediate embedding of a generative model also provides a strong audio representation for downstream classification. Various forms of weak supervision, such as user interaction statistics and visual cues, have also been examined in [24–26].

Our work focuses on developing a recipe of cross-modal supervision using an abundance of text annotations that are weakly associated with the music audio. We benchmark the transfer learning capabilities of the learned representations against analogous past work, and also evaluate different audio encoder architectures.

Cross-modal contrastive learning. Spurred by the success of using contrastive learning to align image features and free-form natural language using large-scale data [1, 2], tri-modal architectures were proposed in [27] and [5] where an audio tower was introduced to the image-text model and contrastive learning is used to enforce the cross-modal alignment. Along the same line in the audio domain, [8] used contrastive learning to align the latent representation of audio and associated tags. The tags come from a fixed vocabulary of size 1K from Freesound [28], and the input to the text encoder was the multi-hot encoded tags. Follow up work in [9] uses a pretrained, non-contextual word embedding (Word2Vec) model to support generalization to new terms beyond the 1K tags. However, this still does not support generalization to free-form natural language. Contrastive learning was also explored in [29] for zero-shot audio classification, using AudioSet and ESC-50 [30] data. Our method focuses on mining a much larger scale collection of audio-text pairs specifically for the music domain. Our data scale supports using state-of-the-art Transformer-based audio and contextual language encoders, which led to a truly arbitrary zero-shot music tagging and retrieval for the first time.

Music text joint embedding models. Content-based music information retrieval requires linking the rich semantics expressible to free-form text with both broad and fine-grained musical properties. One approach is to consider a large number of text label classes and try to ground the semantics in music with a multi-label classification task. In [25], a large vocabulary of 100K n-grams was mined from noisy natural language text associated with music videos. Then, a cross entropy loss was employed to train the music audio encoder, where the softmax layer weights served as text label embeddings that were aligned with audio features by construction. The work in [31] explored various training tasks (classification, regression, metric learning) to align free-form text and music audio, relying on pre-existing emotion labels to connect the modalities.

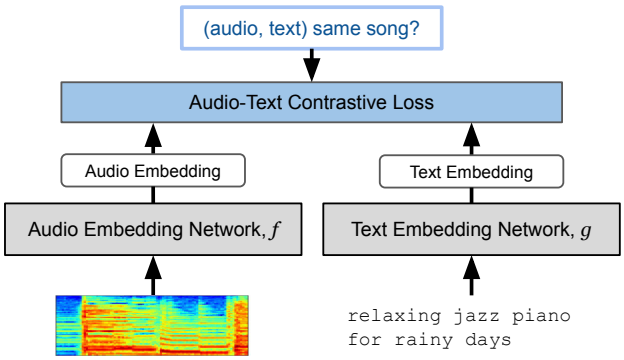

Figure 1. Learning framework diagram.

Closest to our work is MuLaP [32], where 250K audio-caption pairs were mined from a private production music library and used to train a multimodal Transformer with early fusion of the two modalities. Their choice of early fusion, as accomplished with cross-attention layers, restricts the utility of the resulting embeddings to transfer learning applications. Critically, our two-tower parallel encoder approach results in a joint embedding space that provides a natural language interface to arbitrary music audio. This opens up downstream opportunities for cross-modal retrieval, zero-shot tagging, and language understanding.

3. PROPOSED APPROACH

Our goal is to construct a shared embedding space for music audio and free-form natural language text, in which proximity is predictive of shared semantics both within and across modalities. To accomplish this, we rely on cross-modal contrastive learning and a simple two-tower architecture. This is a highly data-intensive endeavor, which we support by mining a large-scale training dataset of (audio, text) pairs. We describe these components in turn below.

3.1 Learning Framework

Figure 1 shows a high-level schematic of the learning framework. Each MuLan model consists of two separate embedding networks for the audio and text input modalities. These networks share no weights, but each terminates in ℓ_2-normalized embedding spaces with the same dimensionality, d. The audio embedding network, $f : \mathbb{R}^{F \times T} \rightarrow \mathbb{R}^d$, takes as input log mel spectrogram context windows with F mel channels and T frames. The text embedding network, $g : \mathcal{A}^n \rightarrow \mathbb{R}^d$ takes as input a null-padded text token sequence of length n over a token vocabulary $\mathcal{A}$.