ADVERSARIAL-MIDIBERT: SYMBOLIC MUSIC UNDERSTANDING MODEL BASED ON UNBIAS PRE-TRAINING AND MASK FINE-TUNING

Zijian Zhao

School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou, China

ABSTRACT

As an important part of Music Information Retrieval (MIR), Symbolic Music Understanding (SMU) has gained substantial attention, as it can assist musicians and amateurs in learning and creating music. Recently, pre-trained language models have been widely adopted in SMU because the symbolic music shares a huge similarity with natural language, and the pre-trained manner also helps make full use of limited music data. However, the issue of bias, such as sexism, ageism, and racism, has been observed in pre-trained language models, which is attributed to the imbalanced distribution of training data. It also has a significant influence on the performance of downstream tasks, which also happens in SMU. To address this challenge, we propose Adversarial-MidiBERT, a symbolic music understanding model based on Bidirectional Encoder Representations from Transformers (BERT). We introduce an unbiased pre-training method based on adversarial learning to minimize the participation of tokens that lead to biases during training. Furthermore, we propose a mask fine-tuning method to narrow the data gap between pre-training and fine-tuning, which can help the model converge faster and perform better. We evaluate our method on four music understanding tasks, and our approach demonstrates excellent performance in all of them. The code for our model is publicly available at https://github.com/RS2002/Adversarial-MidiBERT.

Index Terms— Music Information Retrieval (MIR), Symbolic Music Understanding (SMU), Adversarial Learning, Bidirectional Encoder Representations from Transformers (BERT)

1. INTRODUCTION

Music Information Retrieval (MIR) plays a crucial role in various fields, such as the recommendation systems in music apps and the AI agents for music creation. With the advancement of computer music, symbolic music, which represents music through a structural sequence of notes, has gained widespread attention because most current music is initially created and recorded using symbolic music formats like MIDI [1]. Symbolic Music Understanding (SMU) has been a key research direction within MIR, aiming to assist musicians and amateurs in learning, teaching, and creating music.

Given the similarity between symbolic music and natural language, language models have been widely used in SMU. For example, the Bidirectional Encoder Representations from Transformers (BERT) [2] model has shown promising performance in SMU [3, 4]. An important factor in the success of current language models, especially Large Language Models (LLMs), is their use of large amounts of unlabeled data to pre-train the models to learn basic data structure and relationships. This pre-training mechanism has also been effective in domains with limited data, such as music [5] and signals [6], which can help improve model performance in downstream tasks [7].

Currently, the most popular pre-training method in language model is Mask Language Model (MLM) [2], which is also widely used in pre-trained SMU models [3, 4, 5]. However, due to dataset imbalances, it can lead to bias problems like discrimination in sexism, ageism, and racism in the field of Natural Language Processing (NLP) [8, 9]. For example, in the sentence "She is good at math.", the MLM model would randomly mask some tokens and train the model to recover them. If the masked sentence becomes "[MASK] is good at math.", the model can only recover it according to the training data distribution to achieve the highest accuracy, instead of considering the contextual relationship, because there is no information pointing to the subject. When the training set data distribution about gender in different context scenarios is imbalanced, the model may prefer to recover the [MASK] as "He", resulting in a gender bias problem.

Recently, some studies have indicated that the bias problem can also significantly harm the model performance in downstream tasks including classification and generation [9]. The MLM-based pre-trained models in other areas also suffer from the same problem since the context-free tokens can only be recovered by the data distribution. However, most current methods to solve this problem are limited to the field of NLP and are difficult to transfer to other areas. Most of these methods can only solve single bias problems like sexism [10] and region [11], but other areas do not have a similar concept. For example, data augmentation [12] is a promising method in NLP, but due to the bias problem in other fields like music and signal not being as clear as in natural language, we do not know how to effectively clean, generate or modify the
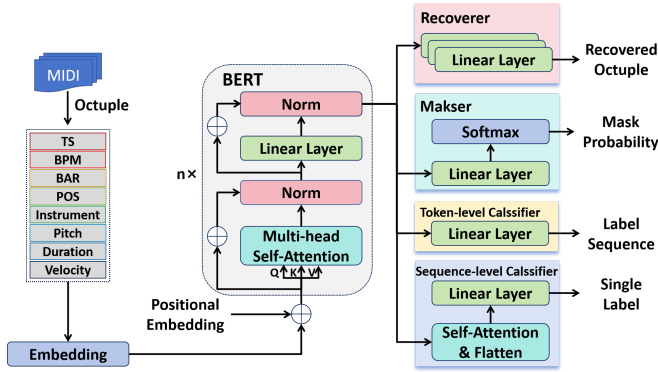
Fig. 1. Network Architecture

data without sufficient domain-specific knowledge.

To address the bias problem in SMU, we first need to consider what we want the model to learn during MLM pre-training. It should be the music structure, relationships, and regularities like basic mode, riff regularities, and modulation regularities, rather than the harmony, chord, or melody direction within specific styles, where those domain-based knowledge would also influence the model performance when the test set has a significant domain gap with the training set. In other words, we want the model to acquire context-dependent information rather than relying solely on the training set's data distribution.

Based on this premise, we propose Adversarial-MidiBERT for SMU: (1) We design an adversarial mechanism to try avoiding masking those context-free tokens to mitigate the bias problem. (2) We propose a mask fine-tuning method that adds random [MASK] tokens during fine-tuning to narrow the data gap between pre-training and fine-tuning. This approach efficiently improves the convergence speed and model performance. (3) Experimental results show that our method achieves excellent performance on four music understanding tasks, including composer classification, emotion classification, velocity prediction, and melody extraction.

2. PROPOSED METHOD

2.1. Overview

The main structure of our Adversarial-MidiBERT is illustrated in Fig. 1. It takes BERT [2] (the encoder part of the Transformer [13]) as the backbone, whose bi-directional attention mechanism can efficiently capture the relationships in music. To adapt BERT for the SMU task, we modify the bottom embedding layer to encode music information and the top output heads for pre-training and downstream tasks.

To embed MIDI music information, we first employ Octuple [3] to represent the symbolic music structure. It transfers each MIDI file into a sequence of tokens, where each token has eight attributes: time signature (TS), tempo (BPM), bar

position (BAR), relative position within each bar (POS), instrument, pitch, duration, and velocity. We then use eight embedding layers to encode these eight attributes respectively and concatenate them together.

As for the top layer of our model, there are four different heads: masker, recoverer, token-level classifier, and sequence-level classifier, all of which share the same backbone. During pre-training, we use the masker to generate the probability of masking each token, and the recoverer would recover the masked tokens. During fine-tuning, we use the classifier heads for classification tasks. The sequence-level classifier generates a single label for a whole music piece, useful for tasks like composer classification and emotion classification. The token-level classifier generates a label sequence corresponding to each token, useful for tasks like velocity prediction and melody extraction.

To address the bias problem in pre-training, we design an adversarial learning mechanism. The masker tries to mask tokens that are difficult for the recoverer to recover, while the recoverer tries to recover all masked tokens. After several epochs, the masker selects context-free tokens with the highest probability, as they can only be inferred according to the training data distribution, leading to the lowest recovery accuracy. We then freeze these tokens to prevent the recoverer from masking them in subsequent epochs. We also apply an unfreezing mechanism, randomly unfreezing some frozen tokens to avoid incorrect freezing. This adversarial process continues until the model converges.

During fine-tuning, we design a mask fine-tuning mechanism, where random [MASK] tokens replace input tokens to reduce the gap between pre-training and fine-tuning. This approach improves convergence speed and model performance.

2.2. Unbias Pre-train

The pre-training process is shown in Fig. 2. First, we perform random transposition to expand the training data, as music datasets are limited. The transposition operation randomly raises or lowers the entire pitch according to the twelve-tone equal temperament within an octave. The transposition range limitation ensures that the style or emotion of the song is not significantly changed by the shift in pitch register. After that, we convert the MIDI file to an Octuple token sequence as the model input.

Within each epoch, the masker first generates the masking probability of each token, and the tokens with the highest $p\%$ masking probabilities are chosen. We follow a similar method to BERT, using the [MASK] token to replace 80% of the chosen tokens and random tokens to replace the remaining 20%. The masked Octuple sequence is then input to the recoverer. We can calculate recovery loss of each masked token accord-
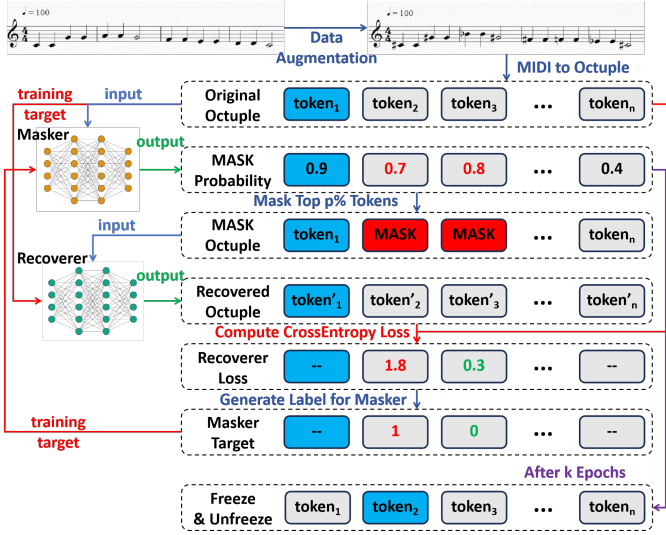
Fig. 2. Pre-train Process: The blue tokens represent the frozen tokens, which cannot be selected as [MASK] tokens.

ing to the following equation:

$$L_i = \sum_{j=1}^{8} w_j \text{CrossEntropy}(\hat{x}_{i,j}, x_{i,j}) \,,$$
$$L_{recoverer} = \sum_{i \in S} L_i \,, \tag{1}$$

where $x_{i,j}$ represents the jth attribute of the ith token, $\hat{x}$ represents the recovered token, w_j is the weight of the jth attribute, and S is the set of masked token indices. The recoverer's loss value is the sum of the recovery loss for those masked tokens. We notice that different attributes have varying convergence speeds and performance, so we design a dynamic weight to balance the loss between them. At the beginning of training, $w_1 \sim w_8$ are set equally to 0.125. Then, in the nth epoch, w_j is set as:

$$w_j = \frac{\frac{1}{a_j}}{\sum_{i=1}^{8} \frac{1}{a_i}} \,, \tag{2}$$

where a_i is the average recovered accuracy of the ith attribute in the $(n-1)^{th}$ epoch. This way, the recoverer pays more attention to the attributes with lower accuracy.

The recovery loss of each token is also used to generate the learning target of the masker, which aims to lower the recovered accuracy of the recoverer by selecting tokens with high loss values. To achieve this, we set the learning target of the top $q\%$ tokens with the highest loss values as 1 and the top $q\%$ tokens with the lowest loss values as 0. The loss function of the masker can be represented as:

$$L_{masker} = \sum_{i \in I_0} \text{MSE}(p_i, 0) + \sum_{i \in I_1} \text{MSE}(p_i, 1) \,, \tag{3}$$

Table 1. Model Configurations

Configuration	Our Setting
Input Length	1024
Network Layers	12
Hidden Size	768
Inner Linear Size	3072
Attn. Heads	12
Dropout Rate	0.1
Optimizer	AdamW
Learning Rate	10^{-4} (pre-train), 10^{-5} (fine-tune)
Batch Size	8
Parameters Mentioned in Section 2 (p, q, a, b, k)	(15,30,30,10,15)
Total Number of Parameters	115 Million

where p_i is the masking probability generated by the masker for the ith token, and I_0, I_1 represent the token index sets with targets set to 0 or 1, respectively.

After repeating this process for k epochs, we believe the tokens with the highest masking probabilities are the most challenging to recover. These tokens correspond to context-free tokens, as they can only be predicted based on the data distribution of the training set, leading to the lowest accuracy. As a result, we freeze the top $a\%$ tokens within each song to avoid them being chosen in the subsequent training, which can be realized by maintaining a dictionary. Simultaneously, we also randomly unfreeze $b\%$ of the frozen tokens to prevent incorrect freezing in the previous step.

2.3. MASK Fine-tune

During fine-tuning, we can still utilize the data augmentation methods employed in pre-training if the downstream tasks are tonality-independent. However, a potential gap may arise since the [MASK] token is present in every epoch during pre-training but is absent in fine-tuning. To address this, we randomly replace $p\%$ of the input tokens with the [MASK] token during fine-tuning. This approach is also similar to the dropout mechanism, which can also help mitigate overfitting.

3. EXPERIMENT

3.1. Experiment Setup

Our model configuration is shown in Table 1. We conduct our experiment using two NVIDIA V100 GPUs. During training, we observe that our Adversarial-MidiBERT occupies about 27GB GPU memory.

The dataset used in this paper is shown in Table 2. We use five public MIDI datasets to train our model. We then conduct four different downstream tasks to evaluate our model's performance, including two token-level classification tasks and two sequence-level tasks:

- Composer Classification: Similar to style classification, composer classification is a more challenging and fine-

Table 2. Dataset Decription

Dataset	Pieces	Task	Task Level	Class Number	Used in Pre-training
ASAP [14]	1068	–	–	–	Y
Pop1K7 [15]	1747	–	–	–	Y
Pianist8 [16]	865	Composer Classification	Sequence Level	8	Y
EMOPIA [17]	1078	Emotion Recognition	Sequence Level	4	Y
POP909 [18]	909	Melody Extraction	Token Level	3	Y
GiantMIDI [19]	10855	Velocity Prediction	Token Level	6	N

Table 3. Model Performance in Different Tasks: The bold and underlined value indicates the best and second best result within each task.

Model	Pre-train			Sequence-Level Classification		Token-Level Classification	
	Accuracy	Epochs	Time	Composer	Emotion	Velocity	Melody
MidiBERT [4]	79.60%	500	6.44d	79.07%	67.59%	44.88%	92.53%
MusicBERT-QM [20]	80.57%	500	9.47d	83.72%	69.52%	46.71%	92.64%
MusicBERT [3]	76.01%	500	10.06d	86.05%	71.06%	38.79%	92.47%
PianoBART [5]	**96.67%**	**268**	**3.19d**	88.37%	73.15%	**49.37%**	92.62%
Adversarial-MidiBERT (ours)	81.47%	436	9.82d	**97.92%**	**79.46%**	45.58%	**92.68%**
Adversarial-MidiBERT (fine-tune w/o mask)	81.47%	436	9.82d	65.98%	70.53%	45.30%	92.55%
Adversarial-MidiBERT (w/o pre-train)	–	–	–	79.76%	68.75%	38.70%	87.98%

grained task. It requires the model to identify which composer created the songs.

- Emotion Recognition: The music emotions in EMOPIA [17] are divided into four types: HVHA, HVLA, LVHA, and LVLA. This task requires the model to classify each song into one of these types.
- Melody Extraction: Each song has different sections, including melody, bridge, and accompaniment. This task requires the model to identify which paragraph each token belongs to.
- Velocity Prediction: Since many MIDI files do not include velocity information, it is important to predict the velocity. We divide velocity into six types and train the model to predict it. To avoid information leakage, we use GiantMIDI [19], which does not participate in pre-training and has its velocity information masked. Since our device could not support training using the full dataset, we select only the first 1000 pieces for the experiment.

We split each dataset into 80% training set, 10% validation set, and 10% testing set. We employ the same early stopping strategy as previous works [5, 4], where the training would stop if the model's accuracy does not increase on the validation set for 30 consecutive epochs. We also set the maximum training epochs to 500.

3.2. Experiment Result

In this section, we compare our method with other SMU models, including MidiBERT [4], MusicBERT [3], MusicBERT-QM [20], and PianoBART [5]. The main differences between these methods lie in their pre-training approaches. For fairness, these BERT-based models use the same backbone as our method. The experimental results are shown in Table

3. It can be seen that our method outperforms the previous BERT-based methods in most tasks, but loses to PianoBART in pre-training and velocity prediction. This may be influenced by the model structure, as the auto-regressive mechanism of the decoder structure in Bidirectional and Auto-Regressive Transformers (BART) [21] makes it more suitable for sequence-level tasks. However, it can be noticed that our model has an extremely significant increase in performance on sequence-level tasks.

We also conducted an ablation study to illustrate our method's performance without pre-training and without masking during fine-tuning. The results show that the model's performance decreases to varying degrees in these cases, demonstrating the effectiveness of our proposed mechanisms. Additionally, we observe that our model has a faster convergence speed than others. By the end of the first epoch, our method achieves relatively high accuracy in all downstream tasks.

4. CONCLUSION

In this paper, we present Adversarial-MidiBERT for SMU, the first method to address the bias problem of pre-trained models in MIR through adversarial pre-training method. Additionally, we introduce a mask fine-tuning approach that significantly enhances the model's accuracy and convergence speed on downstream tasks. Our method achieves remarkable performance on four SMU tasks, especially on the sequence-level tasks. In the future, we aim to explore the application of our method to music generation tasks and NLP tasks.

5. REFERENCES

[1] Joseph Rothstein, *MIDI: A comprehensive introduction*, vol. 7, AR Editions, Inc., 1995.

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, "BERT: Pre-training of deep bidirectional Transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[3] Mingliang Zeng, Xu Tan, Rui Wang, Zeqian Ju, Tao Qin, and Tie-Yan Liu, "Musicbert: Symbolic music understanding with large-scale pre-training," in *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, 2021, pp. 791–800.

[4] Yi-Hui Chou, I Chen, Chin-Jui Chang, Joann Ching, Yi-Hsuan Yang, et al., "Midibert-piano: large-scale pre-training for symbolic music understanding," *arXiv preprint arXiv:2107.05223*, 2021.

[5] Xiao Liang, Zijian Zhao, Weichao Zeng, Yutong He, Fupeng He, Yiyi Wang, and Chengying Gao, "Pianobart: Symbolic piano music generation and understanding with large-scale pre-training," *arXiv preprint arXiv:2407.03361*, 2024.

[6] Zijian Zhao, Tingwei Chen, Fanyi Meng, Hang Li, Xiaoyang Li, and Guangxu Zhu, "Finding the missing data: A bert-inspired approach against package loss in wireless sensing," *arXiv preprint arXiv:2403.12400*, 2024.

[7] Sinong Wang, Madian Khabsa, and Hao Ma, "To pretrain or not to pretrain: Examining the benefits of pretraining on resource rich tasks," *arXiv preprint arXiv:2006.08671*, 2020.

[8] Ismael Garrido-Muñoz, Arturo Montejo-Ráez, Fernando Martínez-Santiago, and L Alfonso Ureña-López, "A survey on bias in deep nlp," *Applied Sciences*, vol. 11, no. 7, pp. 3184, 2021.

[9] Isabel O Gallegos, Ryan A Rossi, Joe Barrow, Md Mehrab Tanjim, Sungchul Kim, Franck Dernoncourt, Tong Yu, Ruiyi Zhang, and Nesreen K Ahmed, "Bias and fairness in large language models: A survey," *Computational Linguistics*, pp. 1–79, 2024.

[10] Shikha Bordia and Samuel R Bowman, "Identifying and reducing gender bias in word-level language models," *arXiv preprint arXiv:1904.03035*, 2019.

[11] Yizhi Li, Ge Zhang, Bohao Yang, Chenghua Lin, Shi Wang, Anton Ragni, and Jie Fu, "Herb: Measuring hierarchical regional bias in pre-trained language models," *arXiv preprint arXiv:2211.02882*, 2022.

[12] Kaiji Lu, Piotr Mardziel, Fangjing Wu, Preetam Amancharla, and Anupam Datta, "Gender bias in neural natural language processing," *Logic, language, and security: essays dedicated to Andre Scedrov on the occasion of his 65th birthday*, pp. 189–202, 2020.

[13] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[14] Francesco Foscarin, Andrew Mcleod, Philippe Rigaux, Florent Jacquemard, and Masahiko Sakai, "Asap: a dataset of aligned scores and performances for piano transcription," in *International Society for Music Information Retrieval Conference*, 2020, number CONF, pp. 534–541.

[15] Wen-Yi Hsiao, Jen-Yu Liu, Yin-Cheng Yeh, and Yi-Hsuan Yang, "Compound word transformer: Learning to compose full-song music over dynamic directed hypergraphs," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021, vol. 35, pp. 178–186.

[16] joann8512, "joann8512/pianist8: First release of pianist8," July 2021.

[17] Hsiao-Tzu Hung, Joann Ching, Seungheon Doh, Nabin Kim, Juhan Nam, and Yi-Hsuan Yang, "Emopia: A multi-modal pop piano dataset for emotion recognition and emotion-based music generation," in *International Society for Music Information Retrieval Conference, IS-MIR 2021*. International Society for Music Information Retrieval, 2021.

[18] Ziyu Wang, Ke Chen, Junyan Jiang, Yiyi Zhang, Maoran Xu, Shuqi Dai, Xianbin Gu, and Gus Xia, "Pop909: A pop-song dataset for music arrangement generation," *arXiv preprint arXiv:2008.07142*, 2020.

[19] Qiuqiang Kong, Bochen Li, Jitong Chen, and Yuxuan Wang, "Giantmidi-piano: A large-scale midi dataset for classical piano music," 2022.

[20] Zhexu Shen, Liang Yang, Zhihan Yang, and Hongfei Lin, "More than simply masking: Exploring pre-training strategies for symbolic music understanding," in *Proceedings of the 2023 ACM International Conference on Multimedia Retrieval*, 2023, pp. 540–544.

[21] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer, "Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," *arXiv preprint arXiv:1910.13461*, 2019.