Expert Systems With Applications 292 (2025) 128613

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Type-based mixture of experts and semi-supervised multi-task pre-training for symbolic music

Shuyu Li [a], Yunsick Sung [b,*]

[a] Department of Multimedia Engineering, Graduate School, Dongguk University-Seoul, Seoul, 04620, Republic of Korea
[b] Department of Computer Science and Artificial Intelligence, Dongguk University-Seoul, Seoul, 04620, Republic of Korea

ARTICLE INFO

Keywords:
Symbolic music
Semi-supervised learning
Multi-task
Mixture of experts
Pre-training
Fine-tuning

ABSTRACT

In the rapidly evolving field of AI-driven music applications, there is a growing interest in the understanding and generation of symbolic music (e.g., MIDI). Symbolic music, unlike audio waveforms, contains discrete representations of musical elements, making it both a detailed and challenging domain for AI models to process. While pre-training techniques from natural language processing have been adapted for music-related tasks, these pre-trained models often struggle with the hierarchical and polyphonic characteristics of symbolic music. To overcome these problems, a method is proposed comprising two components, a foundational model named type-based mixture of experts (TypeMoE) and a semi-supervised multi-task pre-training (SS-MTP) strategy. TypeMoE captures fine-grained musical features more effectively by dynamically activating specialized experts for different event types, while SS-MTP covers tasks including key-signature recognition, time-signature recognition, and causal language modeling. Unlike purely self-supervised approaches, SS-MTP utilizes a small amount of labeled data alongside extensive unlabeled data, enabling structural representation learning and promoting efficient knowledge sharing across tasks. Experimental results showed that TypeMoE, when pre-trained with the SS-MTP strategy, outperformed baseline models in both music understanding and generation tasks. Specifically, it achieved 71.80 % accuracy in genre classification and 76.79 % in emotion classification. For music generation, it outperformed baselines with 54.24 % Hits@1 and 0.7521 BLEU-2 in continue generation, and 75.79 % Hits@1 and 0.8757 BLEU-2 in conditional generation. Additionally, it obtained a CLAP-based semantic alignment score of 0.24.

1. Introduction

Early research on symbolic music (e.g., Musical Instrument Digital Interface (MIDI)) focused on training models from scratch, yielding approaches for music understanding (Hirai & Sawada, 2019; Liang et al., 2020; Madjiheurem, Qu, & Walder, 2016; Salamon & Gómez, 2012; Wang, Li, & Sung, 2023) and music generation (Chu, Urtasun, & Fidler, 2016; Hao-Wen, Dong, Hsiao, Yang, & Yang, 2018; Mao, Shin, & Cottrell, 2018; Yang, Chou, & Yang, 2017). However, with the introduction of Transformer (Vaswani et al., 2017) in natural language processing and its growing application in music-related tasks (Ens & Pasquier, 2020; Huang et al., 2018; Huang & Yang, 2020; Jin et al., 2022; Yu et al., 2022), pre-training on large-scale unlabeled symbolic music data has gained increasing attention. As a widely used format in symbolic music, MIDI provides an ideal foundation for large-scale pre-training due to its expressive and interpretable characteristic. In this format, music is represented as standardized event sequences, which can be tokenized to tokens, enabling models to directly capture patterns in pitch, duration, velocity, and other musical attributes. By leveraging massive amounts of unlabeled data, pre-trained models can learn inherent musical patterns and acquire general-purpose representations that are transferable across multiple musical tasks.

Recent studies have applied large-scale pre-trained models to symbolic music understanding and generation, closely mirroring approaches in natural language processing. LakhNES (Donahue et al., 2019) is utilized to generate multi-instrumental music. MusicBERT (Zeng et al., 2021) and MidiBERT (Chou et al., 2024) have achieved significant progress in various musical understanding tasks, including melody completion, melody extraction, velocity prediction, accompaniment suggestion, genre classification, emotion classification, and style classification. PianoBART (Liang et al., 2024) has further advanced tasks such as music continuation, velocity prediction, melody extraction, emotion recognition, and composer classification. Lastly, MuPT (Qu et al., 2024) has improved the quality of generated music.

[*] Corresponding author.
E-mail address: sung@dongguk.edu (Y. Sung).

https://doi.org/10.1016/j.eswa.2025.128613
Received 11 February 2025; Received in revised form 29 April 2025; Accepted 12 June 2025
Available online 17 June 2025
0957-4174/© 2025 Elsevier Ltd. All rights reserved, including those for text and data mining, AI training, and similar technologies.

Despite significant progress in recent years, most traditional approaches focused on either understanding or generation tasks. Motivated by this observation, this paper proposes a novel foundational model along with a refined pre-training strategy, aiming to consider both understanding and generation tasks.

During foundation model development, several architectural characteristics were considered. The causal attention matrix utilized in the decoder-only architecture, with its lower-triangular structure, inherently maintains full rank, thus offering superior modeling capacity compared to bidirectional matrices of the encoder-only architecture prone to low-rank degradation (Dong, Jean-Baptiste, Cordonnier, & Loukas, 2021). Token prediction in the decoder-only architecture accesses less contextual information at each position, increasing prediction difficulty; yet sufficient model scale and training data convert this challenge into enhanced generalized representation learning (Radford, Wu, Child, Luan, & Amodei, 2019). The decoder-only architecture also facilitates prompt interaction at each decoder layer, significantly improving few-shot adaptation and fine-tuning effectiveness (Brown et al., 2020). Additionally, causal attention inherently encodes positional information, mitigating the position-invariance issues in the encoder-based architecture, which may yield identical representations despite token reordering (Haviv et al., 2022). Moreover, efficient reuse of key-value caches for accelerated inference is naturally supported, a feature comparatively challenging in the encoder-decoder architecture. Given these advantages, the foundational model proposed in this paper adopts the decoder-only architecture.

To effectively capture fine-grained musical features within a decoder-only architecture, the Mixture of Experts (MoE) mechanism (Fedus, Zoph, & Shazeer, 2022) is referenced. Standard MoE employs a gating network to dynamically route inputs to specialized experts, maintaining model capacity with reduced computational overhead. However, standard MoE bases routing decisions on token embeddings, assigning semantically similar events to the same experts. Although given MIDI events possess distinct types and functionalities, similarity-based routing may still group events with divergent processing needs, leading to several limitations. Standard MoE does not inherently distinguish between different event types, potentially mixing unrelated musical functions. In addition, relying only on semantic similarity might overlook subtle connections among musical attributes. Moreover, the gating network also tends to favor frequent events, often overlooking rare but important control events. To address these limitations, a routing mechanism based on type embeddings is proposed, enabling precise expert specialization for diverse musical event types and enhancing fine-grained feature learning. And since type embeddings typically have lower dimensionality than event embeddings, computational overhead is further reduced. This novel foundation model is termed Type-based Mixture of Experts (TypeMoE).

Causal language modeling (CLM), a widely adopted pre-training strategy in the decoder-only architecture, was used as a reference for designing a suitable pre-training strategy for TypeMoE. CLM employs an autoregressive objective wherein the model learns to predict subsequent tokens conditioned solely on preceding context. Several prior studies have explored incorporating meta-information, such as key- and time-signatures, to facilitate a deeper understanding of musical structure. In MusicBERT (Zeng et al., 2021), the time-signature is embedded as one part of the input tokens, whereas MuPT (Qu et al., 2024) explicitly incorporates both key- and time-signatures into the input sequence. However, directly embedding into the input may cause the model to overly rely on structural cues, potentially hindering its capacity for autonomous feature learning and generalization. To facilitate the learning of structural representation, key- and time-signatures can be employed as prediction targets instead of input tokens, thereby enhancing the capability of the model to infer and understand musical structures autonomously. This novel pre-training strategy is termed semi-supervised multi-task pre-training (SS-MTP), which is motivated by the observation that time-signatures can be inferred from rhythmic patterns involving note positions and durations, whereas key-signatures can be derived through pitch-based analysis.

In summary, this paper proposes a method integrating two components: a Type-based Mixture of Experts (TypeMoE) foundation model and a semi-supervised multi-task pre-training (SS-MTP) strategy. The proposed method aims to effectively capture both fine-grained and structural representations of symbolic music. Compared to standard MoE, TypeMoE effectively accommodates the multi-type nature of symbolic music by assigning dedicated experts to distinct event types. This design mitigates interference among experts and improves feature extraction, contrasting with the gating approach of the standard MoE, wherein all events compete within a shared representational space. By decomposing the gating mechanism according to event types, experts can specialize in capturing distinct musical characteristics. Furthermore, this type-driven mechanism enhances interpretability and contributes to more stable training dynamics; while the SS-MTP strategy integrates key-signature recognition (KSR), time-signature recognition (TSR), and CLM within a unified framework, effectively exploiting both labeled and unlabeled data to facilitate robust representation learning. By sharing parameters and transferring information across related tasks, SS-MTP captures both contextual and structural musical features, thereby promoting cross-task knowledge transfer. The learned representations transcend task-specific optimization, offering a generalizable and versatile foundation for diverse downstream applications spanning music understanding and generation.

Additionally, four downstream tasks are designed for fine-tuning the pre-trained model and systematically evaluating effectiveness in symbolic music understanding and generation. The classification tasks, including genre classification and emotion classification, measure the capability of recognizing and distinguishing musical features, whereas the generation tasks, including continue generation and conditional generation, assess compositional capability. These downstream tasks provided evaluation using multiple measures, validating the applicability of the proposed model across music understanding and generation domains.

The primary contributions of the proposed method are summarized as follows:

1. By assigning dedicated experts to distinct event types, TypeMoE effectively mitigates interference among experts and strengthens fine-grained feature extraction capabilities.
2. SS-MTP integrates KSR, TSR, and CLM within a unified framework, effectively utilizing labeled and unlabeled data to capture contextual information, learn structural representation, and facilitate cross-task knowledge transfer.
3. Four downstream tasks, including genre classification, emotion classification, continue generation, and conditional generation, are introduced for fine-tuning, providing evaluation using multiple measures for both music understanding and generation capabilities, thereby ensuring adaptability of the proposed model across diverse musical applications.

2. Related work

Recent years have witnessed substantial advancements in symbolic music processing, driven primarily by the widespread adoption of Transformer architecture and advancements in pre-training methodologies. This section provides an analysis and summary of foundation models utilized in symbolic music processing, together with their corresponding pre-training strategies.

2.1. Foundational models

In recent years, Transformer-based models have achieved remarkable advancements in symbolic music processing. Inspired by developments in natural language processing, these models employ analogous approaches for both musical understanding and generation tasks. Typically, these foundation models can be categorized into three main types:

encoder-only, decoder-only, and encoder-decoder models, each serving distinct functions within music understanding and generation tasks. Additionally, this section introduces several recent enhancements to Transformer architecture.

2.1.1. Encoder-only models

The encoder-only architecture is exemplified by BERT (Devlin, Chang, Lee, & Toutanova, 2019), which employs a Transformer encoder trained using MLM to capture bidirectional representations within token sequences. Representative BERT-based studies in symbolic music processing include the following.

For instance, MusicBERT enhances the standard BERT by proposing the Octuple representation along with a bar-level masking strategy (Zeng et al., 2021). Octuple encodes each note using multiple musical attributes, significantly reducing sequence length and improving efficiency in capturing long-range musical contexts. Moreover, the adoption of octuple-level and bar-level masking instead of token-level masking mitigates over-reliance on local cues, thus promoting a deeper understanding of complex musical relationships. Similarly, MidiBERT (Chou et al., 2024) employs the BERT framework and compound word (CP) representation to facilitate symbolic music understanding, leveraging CP-level masking as a self-supervised objective to obtain rich musical representations that can subsequently be fine-tuned across various downstream understanding tasks.

However, due to the reliance on MLM or its variants for capturing bidirectional contexts and the absence of an autoregressive mechanism, this architecture often struggles with generating coherent musical sequences and handling time-sensitive tasks, such as smoothly continuing melodies or composing novel musical content.

2.1.2. Decoder-only models

The decoder-only architecture has demonstrated significant success in symbolic music generation owing to its autoregressive nature. By predicting tokens sequentially from left to right, this architecture naturally supports sequential composition tasks, ensuring musical continuity and thematic coherence over extended time horizons.

A representative example is LakhNES (Donahue et al., 2019), which employs TransformerXL (Dai, 2019) to generate multi-instrumental music. By pre-training on the extensive Lakh MIDI dataset and subsequently fine-tuning on the NES-MDB dataset, LakhNES effectively mitigates the challenge posed by limited data availability for specific musical styles. Additionally, the recurrent mechanism integrated into TransformerXL enables the model to capture long-range dependencies, producing compositions characterized by enhanced coherence and stylistic consistency. Similarly, MuPT (Qu et al., 2024) utilizes Llama2 (Touvron et al., 2023) trained on extended sequences and introduces the SMT-ABC representation, which sustains coherence across multiple instruments by aligning musical measures and preserving structural relationships, thus facilitating structurally consistent music generation.

Although the decoder-only architecture has seldom been applied to music understanding tasks, it has consistently achieved remarkable performance across various natural language understanding benchmarks, even surpassing the encoder-only architecture in certain applications (Achiam et al., 2023; Brown et al., 2020; Chowdhery et al., 2023; Rae et al., 2021). Due to its strong performance in both understanding and generation tasks, the TypeMoE proposed in this paper adopts the decoder-only architecture.

2.1.3. Encoder-decoder models

The encoder-decoder architecture, a widely adopted sequence-to-sequence approach, has recently gained increasing popularity in symbolic music processing. Due to its inherent flexibility, this architecture is particularly suitable for a broad range of applications.

PianoBART (Liang et al., 2024) is an encoder-decoder model explicitly designed for symbolic music generation and understanding. Built upon the BART (Lewis et al., 2019), which is a denoising autoencoder (DAE) for pre-training sequence-to-sequence models, PianoBART utilizes a Transformer encoder to learn bidirectional representations and a decoder for autoregressive output generation. By adopting Octuple encoding, PianoBART efficiently captures long-term dependencies while significantly reducing sequence length. Moreover, a dynamic masking strategy at the n-bar level during pre-training discourages over-reliance on local patterns, encouraging the model to capture more abstract and robust musical relationships. Consequently, PianoBART exhibits strong performance across both symbolic music understanding and generation tasks, benefiting from unified representations learned during pre-training.

Nevertheless, the encoder-decoder architecture continues to face several challenges in symbolic music tasks. The dual-module design requires operating both the encoder and decoder during inference, thereby increasing model complexity and computational overhead. Moreover, the encoder occupies a portion of the model parameters, potentially reducing the parameter capacity available to the decoder for generation tasks, thus limiting overall generation performance.

2.1.4. Enhancements for transformer

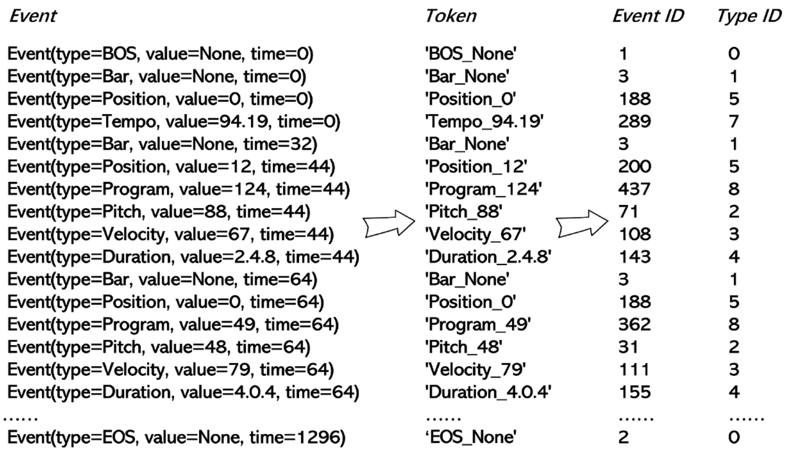
Four key modifications are introduced to the traditional Transformer architecture: grouped-query attention (GQA) (Ainslie et al., 2023), root mean square layer normalization (RMSNorm) (Zhang & Sennrich, 2019), rotary position embedding (RoPE) (Su et al., 2024), and Mixture of Experts (MoE) (Fedus et al., 2022). GQA partitions the query vector into multiple groups, enabling more granular and specialized attention computations within each group, thus improving computational efficiency and effectively capturing nuanced relationships across distinct feature subspaces. RMSNorm replaces standard layer normalization (LayerNorm) with a normalization method based on the root mean square, resulting in more stable training dynamics and improved convergence behavior. RoPE provides continuous and flexible positional encoding, preserving rotational invariance within the feature space and effectively modeling long-range dependencies. MoE scales model capacity by dynamically routing tokens to specialized experts, thereby enhancing expressive power while managing computational overhead. In contrast to traditional expert systems, which typically rely on manually curated rules or domain-specific knowledge suitable primarily for well-defined application domains (Modi et al., 2011; Saibene, Assale, & Giltri, 2021; Walek & Fajmon, 2023), MoE employs multiple sub-networks to automatically specialize in distinct sub-distributions or feature representations, using a routing mechanism to dynamically select the most appropriate expert outputs. This approach allows flexible adaptation to diverse inputs and improves overall performance without requiring manually encoded rules. In this paper, these improvements to the Transformer are combined with a type-based routing mechanism, resulting in a new foundation model called TypeMoE.

2.2. Pre-training strategies

In symbolic music processing, pre-training strategies play a crucial role in improving model performance. The three main strategies are MLM, CLM, and DAE, each having unique features and being suitable for different applications.

2.2.1. Mask language modeling

MLM randomly masks certain tokens within the input sequence, prompting the model to predict these masked tokens based on their surrounding context. Leveraging bidirectional contextual information, MLM enables the model to learn richer representations. Typically, a subset of the input tokens is randomly selected for masking; for instance, in models such as BERT (Zeng et al., 2021), approximately 15 % of tokens are masked. Among these masked tokens, 80 % are replaced with a special [MASK] token, 10 % are substituted with random tokens, and the remaining 10 % remain unchanged. This masking strategy encourages the model not only to predict masked tokens but also to



Event	Token	Event ID	Type ID
Event(type=BOS, value=None, time=0)	'BOS_None'	1	0
Event(type=Bar, value=None, time=0)	'Bar_None'	3	1
Event(type=Position, value=0, time=0)	'Position_0'	188	5
Event(type=Tempo, value=94.19, time=0)	'Tempo_94.19'	289	7
Event(type=Bar, value=None, time=32)	'Bar_None'	3	1
Event(type=Position, value=12, time=44)	'Position_12'	200	5
Event(type=Program, value=124, time=44)	'Program_124'	437	8
Event(type=Pitch, value=88, time=44)	'Pitch_88'	71	2
Event(type=Velocity, value=67, time=44)	'Velocity_67'	108	3
Event(type=Duration, value=2.4.8, time=44)	'Duration_2.4.8'	143	4
Event(type=Bar, value=None, time=64)	'Bar_None'	3	1
Event(type=Position, value=0, time=64)	'Position_0'	188	5
Event(type=Program, value=49, time=64)	'Program_49'	362	8
Event(type=Pitch, value=48, time=64)	'Pitch_48'	31	2
Event(type=Velocity, value=79, time=64)	'Velocity_79'	111	3
Event(type=Duration, value=4.0.4, time=64)	'Duration_4.0.4'	155	4
......	......	......	......
Event(type=EOS, value=None, time=1296)	'EOS_None'	2	0

Fig. 1. Illustration of the MIDI encoding process.

discriminate between original and replaced tokens, thereby enhancing robustness.

Since [MASK] tokens are not employed during inference, a mismatch arises between training and inference conditions. Furthermore, MLM primarily emphasizes predicting masked tokens, rendering it less suitable for purely autoregressive generation tasks.

2.2.2. Causal language modeling

CLM predicts the next token in a sequence exclusively based on preceding context, thereby restricting model access to future information. This causal modeling approach is particularly suitable for generative tasks such as music composition. In symbolic music generation, CLM is commonly adopted within the decoder-only architecture, enabling coherent and contextually consistent musical outputs. Specifically, during CLM training, the input sequence is converted into input-label pairs, where the labels correspond to the input sequence shifted by one position.

However, the causal nature of CLM limits the ability to leverage global information. In this paper, to overcome this limitation, structural prediction tasks, including KSR and TSR, are integrated into the CLM framework. The proposed method represents a form of semi-supervised training. The prefix “semi-” appears broadly across various algorithms and research domains, such as semi-empirical (Etesami, Shirangi, Mehrdad, & Zhang, 2021), semi-parametric (Zhang et al., 2023), and semi-structured (Adeoye-Olatunde & Olenik, 2021), generally indicating partial or hybrid methodologies. Distinct from these concepts, semi-supervised learning specifically combines a small amount of labeled data with a substantially larger pool of unlabeled data, thereby benefiting simultaneously from the clear guidance provided by labeled examples and the richer distributional patterns inherent in unlabeled samples.

2.2.3. Denoising autoencoder

DAE can be employed in pre-training for text generation by introducing random perturbations at the encoder stage and reconstructing the original text autoregressively through the decoder (Lewis et al., 2019). In practice, input sequences may undergo diverse forms of corruption, such as masking continuous spans of tokens, completely removing selected tokens, shuffling sentence order, or permuting entire text segments. Unlike MLM, which typically masks a fixed proportion of individual tokens, DAE emphasizes more substantial perturbations at the sentence or paragraph level, thus providing greater flexibility in reconstructing and generating coherent text.

Because the training objective of DAE differs from the strictly autoregressive nature of generation tasks, it can lead to inconsistencies that may negatively impact performance. Additionally, DAE typically requires substantial data and training resources due to relatively lower data efficiency.

3. Method

TypeMoE and SS-MTP are proposed to enhance symbolic music representations. Evaluation using multiple measures on downstream tasks, including genre classification, emotion classification, continue generation, and conditional generation, demonstrates the effectiveness of the pre-trained model and highlights promising potential for diverse music applications.

3.1. Data representation

This paper adopts REMI+ (Rütte, Biggio, Kilcher, & Hofmann, 2022) as the representation method, which extends the original REMI representation. REvamped MIDI-derived Events (REMI) (Huang & Yang, 2020) encodes MIDI data into discrete tokens, where notes are represented by pitch, duration, and velocity tokens, and temporal information is captured using bar and position tokens. The bar token indicates the beginning of a new bar, whereas the position token specifies the note position within the current bar according to a predefined resolution. REMI+ extends REMI to support multi-track symbolic music by introducing program tokens preceding pitch tokens, allowing the representation of multiple instruments. This extended format facilitates versatile modeling of more complex musical compositions. As illustrated in Fig. 1, MIDI events are extracted, converted into tokens, and subsequently mapped to event and type IDs.

Table 1 summarizes each event type used in the REMI+ representation as follows. The *PAD* token (Type ID = 0) is reserved for sequence padding, while *BOS* and *EOS* tokens (Type ID = 0) respectively mark the beginning and end of sequences. *Bar* token (Type ID = 1) indicates measure boundaries to clarify musical structure. *Pitch* tokens (Type ID = 2) represent MIDI notes ranging from 21 to 108, covering the standard musical pitch range. *Velocity* tokens (Type ID = 3) capture discrete loudness levels from 3 to 127 at intervals of 4, reflecting practical performance dynamics. *Duration* tokens (Type ID = 4) follow the format `Duration_x.y.z`, where x (0–12) denotes the number of full beats, y (0–7) specifies the sub-beat index within each beat, and z indicates the resolution (commonly 4 or 8 subdivisions per beat), thereby encoding note lengths. *Position* tokens (Type ID = 5) specify sub-beat offsets within a bar (0–31), offering fine-grained timing control. *PitchDrum* tokens (Type ID = 6) represent percussion hits covering MIDI note numbers 27 to 87, capturing commonly used drum sounds. *Tempo* tokens (Type ID = 7) span 40.0 to 250.0 BPM in discrete intervals, encompassing a wide range of musical tempos. *Program* tokens (Type ID = 8) encode General MIDI instrument numbers

Table 1
Overview of types, type IDs, event ID ranges, and included tokens for the REMI+ representation. The "Non-MIDI" type is reserved for tokens that lie outside standard MIDI events.

Type	Type ID	Event ID	Tokens included
PAD	0	0	PAD_None
BOS	0	1	BOS_None
EOS	0	2	EOS_None
Bar	1	3	Bar_None
Pitch	2	4–91	Pitch_21 to Pitch_108
Velocity	3	92–123	Velocity_3, Velocity_7, Velocity_11, ..., Velocity_127
Duration	4	124–187	Duration_x.y.z (x = 0–12, y = 0–7, z = 4 or 8)
Position	5	188–219	Position_0 to Position_31
PitchDrum	6	220–280	PitchDrum_27 to PitchDrum_87
Tempo	7	281–312	Tempo_40.0, Tempo_46.77, Tempo_53.55, ..., Tempo_250.0
Program	8	313–441	Program_0 to Program_127, and Program_-1 (Program_-1 for Drum)
Non-MIDI	9	—	—

(0–127) alongside a dedicated *Program_-1* for drums, unifying pitched and percussive instruments. Finally, the *Non-MIDI* type (Type ID = 9) is reserved for tokens that lie outside standard MIDI event definitions, allowing for future extensions beyond the MIDI specification.

3.2. Type-based mixture of experts

As illustrated in Fig. 2, the model architecture comprises an Event Embedding Layer, a Type Embedding Layer, and a stack of decoder blocks. TypeMoE accepts two types of inputs: `Event IDs` and their corresponding `Type IDs`. These inputs are respectively mapped into event hidden states h and type hidden states t through Event Embedding Layer and Type Embedding Layer.

During the processing of h, RMSNorm is applied to enhance training stability, after which the normalized output is passed to the GQA Layer. Within the GQA Layer, the normalized hidden states h are projected into query q, key k, and value v vectors, followed by the application of RoPE to the q and k vectors. Subsequently, the attention output a is computed via grouped-query attention based on q, k, and v. Finally, the attention output a is combined with the original hidden states h through a residual connection, yielding the updated hidden states h as follows:

$$\begin{aligned}
&h = \mathrm{RMSNorm}(h),\\
&q = W_q h, \quad k = W_k h, \quad v = W_v h,\\
&q = \mathrm{RoPE}(q), \quad k = \mathrm{RoPE}(k), \qquad (1)\\
&a = \mathrm{GQA}(q, k, v),\\
&h = a + h.
\end{aligned}$$

where W_q, W_k, and W_v denote the projection weights for the query, key, and value, respectively.

Meanwhile, t is processed by RMSNorm and subsequently passed through the Type Projection Layer. The projected t is then fed into Router to obtain routing scores s. Through the Top-k operation, the highest k scores, denoted as $s_{\text{Top-}k}$, along with their corresponding indices $i_{\text{Top-}k}$, are extracted. Finally, a softmax operation is applied to $s_{\text{Top-}k}$ to derive normalized weights w as follows:

$$\begin{aligned}
t &= \mathrm{RMSNorm}(t),\\
t &= tW_t, \quad W_t \in \mathbb{R}^{d\times d},\\
s &= tW_r, \quad W_r \in \mathbb{R}^{d\times n}, \qquad (2)\\
s_{\text{Top-}k}, i_{\text{Top-}k} &= \text{Top-}k(s, k),\\
w &= \mathrm{softmax}\left(s_{\text{Top-}k}\right).
\end{aligned}$$

where n denotes the total number of experts, and k indicates the number of selected experts. W_t represents the weights of the Type Projection Layer, and W_r denotes the weights of Router.

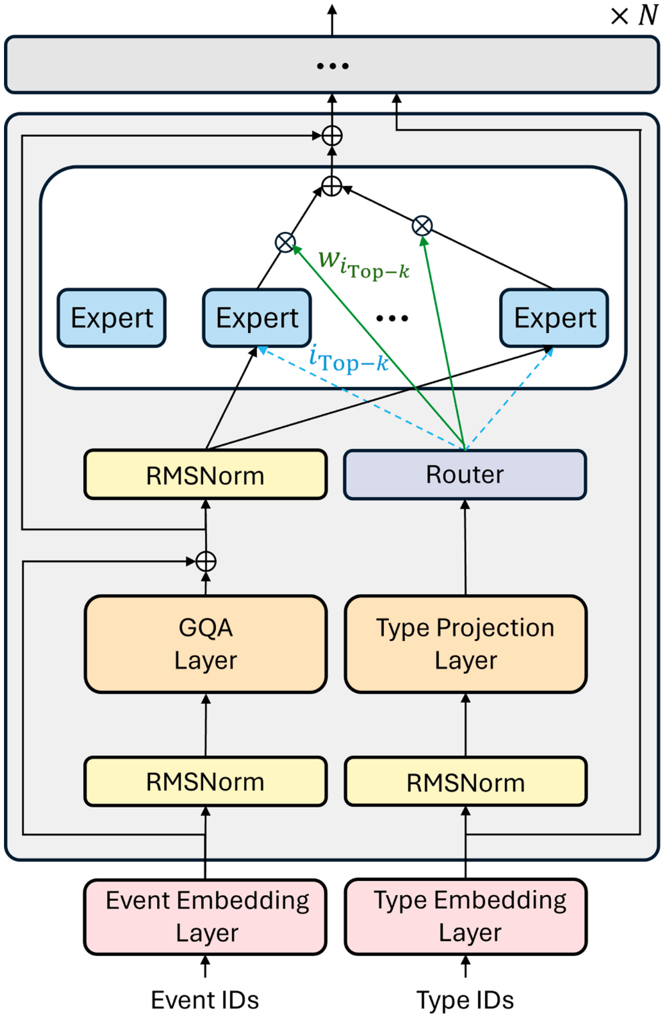


Fig. 2. Illustration of the proposed TypeMoE architecture.

The indices $i_{\text{Top-}k}$ are utilized to select the corresponding experts. Each selected expert $f_e(\cdot)$ then operates on the hidden states h, which have been normalized by RMSNorm, as follows:

$$o_e = f_e(h), \quad e \in i_{\text{Top-}k}. \qquad (3)$$

where e indicates the index of the e-th selected expert. The outputs o_e from all activated experts are aggregated into a final output o through a weighted sum, defined as:

$$o = \sum_{e \in i_{\text{Top-}k}} w_e\, o_e. \qquad (4)$$

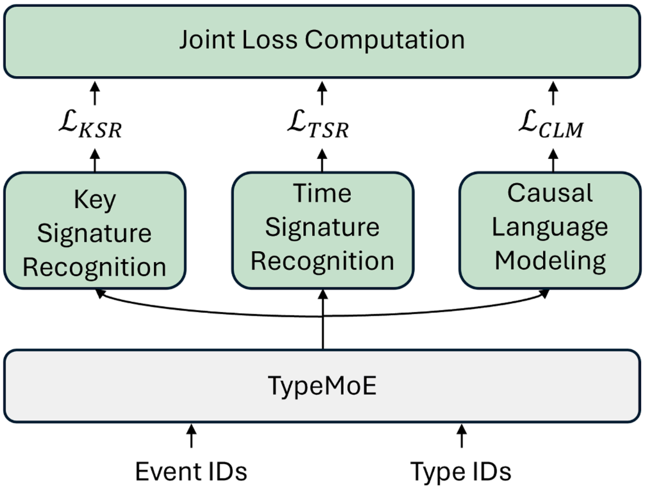


Fig. 3. Illustration of the SS-MTP pipeline.

3.3. Semi-supervised multi-task pre-training

SS-MTP integrates KSR, TSR, and CLM, allowing the model to learn the structural patterns in musical data, as illustrated in Fig. 3. Joint learning of these tasks enables TypeMoE to capture richer musical features, thus enhancing generalization and deepening structural understanding. Key- and time-signatures, readily available in music datasets, serve as high-quality annotations to further refine the representation learning process. Specifically, KSR identifies key-signatures, such as *C major* or *G major*, helping the model capture tonal structures effectively. Meanwhile, TSR determines time-signatures, such as *4/4* or *3/4*, enabling the model to better represent rhythmic and timing information. Lastly, CLM serves as an autoregressive generative task, training the model to predict subsequent events in sequences, thereby facilitating the learning of fundamental musical structures and contextual relationships.

During pre-training, the final decoder output o is employed to generate predictions for CLM, KSR, and TSR. Each task has a dedicated output head that processes o to produce the predicted probability distribution $\hat{y}$. Specifically, for a task $T \in \{\text{CLM}, \text{KSR}, \text{TSR}\}$, $\hat{y}_T$ denotes the predicted probability distribution, while y_T represents the ground-truth distribution. The cross-entropy losses and the overall joint loss are defined as follows:

$$\begin{aligned} L_T &= -\, y_T \,\log\left(\hat{y}_T\right), \\ L_{\text{joint}} &= L_{\text{CLM}} + \lambda\left(L_{\text{KSR}} + L_{\text{TSR}}\right). \end{aligned} \tag{5}$$

where λ is a hyperparameter used to balance the primary loss with the two auxiliary losses. This multi-task framework not only enhances causal language modeling capabilities but also facilitates accurate recognition of key- and time-signatures, thereby promoting a structural understanding of music.

3.4. Fine-tuning on musical downstream tasks

During fine-tuning, the pre-trained model undergoes adaptive adjustments for downstream tasks, including genre classification, emotion classification, continue generation, and conditional generation. These downstream tasks are specifically designed for fine-tuning, aiming to enhance model performance in targeted domains and systematically evaluate the capabilities of the pre-trained model. Detailed descriptions of each task are provided as follows.

In the genre classification task, the model aims to accurately identify musical genres (e.g., classical, pop, rock, electronic) by minimizing cross-entropy loss. Let $y \in \mathbb{R}^C$ represent the one-hot encoded ground-truth label distribution across C classes, and $\hat{y} \in \mathbb{R}^C$ denote the predicted probability distribution output by the model. The cross-entropy loss is defined as follows:

$$\mathcal{L} = -\sum_{c=1}^{C} y_c \log\left(\hat{y}_c\right). \tag{6}$$

Emotion classification seeks to identify emotional characteristics embedded within musical tokens, drawing on the widely adopted circumplex model of affect (Russell, 1980), which categorizes affect into four quadrants: HVHA (high valence, high arousal), HVLA (high valence, low arousal), LVHA (low valence, high arousal), and LVLA (low valence, low arousal). Through appropriate parameter adjustments, the model effectively captures subtle emotional nuances. The loss function adopts the same cross-entropy formulation used in the genre classification task.

The continue generation task involves predicting a randomly omitted end segment of an input sequence, conditioned on the observed segment x. Let N denote the length of the omitted segment and C represent the number of token classes in the model vocabulary. The model is required to predict the omitted portion. Suppose the ground-truth one-hot distributions for the missing tokens are represented by $y \in \mathbb{R}^{N\times C}$, and the predicted probability distributions by $\hat{y} \in \mathbb{R}^{N\times C}$. The index n enumerates the N missing tokens, while c enumerates the C classes for each token. The cross-entropy loss is formally computed as follows:

$$\mathcal{L} = -\sum_{n=1}^{N}\sum_{c=1}^{C} y_{n,c}\, \log\left(\hat{y}_{n,c} \mid x\right). \tag{7}$$

For conditional generation, the model integrates a prompt encoder utilizing P-tuning V2 (Liu et al., 2021). Let p denote the encoded prompt, conditioning the entire generation process. Suppose $y \in \mathbb{R}^{N\times C}$ represents the ground-truth label distribution, and $\hat{y} \in \mathbb{R}^{N\times C}$ denotes the predicted probability distribution conditioned on the prompt p, where C is the vocabulary size. Under this setting, the cross-entropy loss is computed by summing over both the sequence length N and the number of classes C, as follows:

$$\mathcal{L} = -\sum_{n=1}^{N}\sum_{c=1}^{C} y_{n,c} \log(\hat{y}_{n,c} \mid p) \tag{8}$$

4. Experiment

In this section, experiments were conducted on understanding and generation tasks. Comparative evaluations with baseline and ablation models demonstrated the effectiveness of integrating key- and time-signatures during pre-training and the benefits of adopting the TypeMoE model.

4.1. Datasets

Several datasets were employed to support pre-training and evaluation across diverse downstream tasks. These datasets were selected based on suitability to specific research objectives, including MidiCaps (Melechovsky, Roy, & Herremans, 2024), POP1K7 (Hsiao, Liu, Yeh, & Yang, 2021), POP909 (Wang et al., 2020), and EMOPIA (Hung et al., 2021).

- **MidiCaps** is a large-scale dataset containing 168,385 MIDI files paired with descriptive textual captions, created through a pipeline integrating MIR-based feature extraction with Claude 3 (Anthropic, 2024) using in-context learning. The MIDI files originate from the Creative Commons-licensed Lakh MIDI Dataset (Raffel, 2016).
- **EMOPIA** is a multi-modal dataset (audio and MIDI) designed to study perceived emotions in pop piano music. It contains 1,087 clips extracted from 387 songs, each annotated at the clip level with emotion labels provided by four annotators, facilitating research on emotion-related music tasks.

Table 2
Summary of datasets employed across pre-training and downstream tasks, including details on the number of musical pieces (Pieces), clips (Clips), total duration in hours (Hours), and their utilization in pre-training and specific downstream tasks (Genre Classification, Emotion Classification, Continue Generation, and Conditional Generation).

Dataset	Pieces	Clips	Hours	Pre-Training	Genre	Emotion	Continue	Conditional
MidiCaps	168,385	N/A	8,985	✓	✓			✓
EMOPIA	387	1,087	11			✓		
Pop1K7	1,748	N/A	108				✓	
POP909	909	N/A	60				✓	

Table 3
Coverage of key musical factors in symbolic music representation approaches. A check mark (✓) indicates that the feature is included, while a cross (×) means it is not.

Representation	Pitch	Position	Bar	Velocity	Duration	Program	Tempo	TimeSignature
CP (MidiBERT)	✓	✓	✓	✓	✓	×	✓	×
Octuple (MusicBERT, PianoBART)	✓	✓	✓	✓	✓	✓	✓	✓
Note with Time-shift (LakhNES)	✓	✓	×	×	✓	✓	×	×
SMT-ABC (MuPT)	✓	×	✓	×	✓	✓	×	✓
REMI+ (TypeMoE)	✓	✓	✓	✓	✓	✓	✓	✓

- **POP1K7** is a dataset consisting of 1,748 pop piano pieces totaling approximately 108 h of music, with an average song duration of four minutes. All pieces share a common 4/4 time-signature. Audio recordings are converted into symbolic sequences using the "Onset and Frames" transcription method and beat-synchronized using the Madmom library.
- **POP909** is a dataset containing piano arrangements of 909 popular songs created by professional musicians. Each song includes MIDI representations of the vocal melody, lead instrument melody, and piano accompaniment, all precisely aligned with the original audio recordings.

The utilization of these datasets across the downstream tasks is summarized in Table 2, detailing the number of pieces, clips, total duration (in hours), and the specific tasks supported by each dataset.

4.2. Data representation

Table 3 compares five symbolic music representation approaches that differ in feature coverage and encoding style. Two-dimensional approaches, such as CP (Chou et al., 2024) and Octuple (Liang et al., 2024; Zeng et al., 2021), bundle multiple attributes into single tokens for compactness, while one-dimensional approaches, such as Note with Time-shift (Donahue et al., 2019), SMT-ABC (Qu et al., 2024), and REMI+, represent each attribute with a separate token. By merging several attributes into one token, two-dimensional approaches reduce the total token count and thus accommodate more music segments within a fixed sequence length, capturing strong interactions among pitch, velocity, duration, and other musical elements. However, a single shared vector may not fully represent each attribute when the number of attributes is large or they vary widely. In contrast, one-dimensional approaches provide separate tokens and hidden states for each attribute, isolating different elements for clearer modeling and reducing conflicts. This design also enables greater control if a task requires direct manipulation of specific attributes, but it increases sequence length and may therefore raise computational demands.

Moreover, the coverage of musical factors varies across these approaches. While some, such as Octuple and REMI+, include many key factors, others omit attributes like velocity, tempo, or time-signature. Balancing sequence length with the desired level of detail is thus crucial for choosing an approach suited to specific task requirements.

4.3. Evaluated models and hyperparameters

Table 4 showed the representation approaches and pre-training strategies used by the evaluated models. TypeMoE and its three ablated versions adopted REMI+ as the representation. The standard version and the variant without type embedding (w/o Type Embedding) both used SS-MTP as the pre-training strategy. The variant without multi-task training (w/o Multi-Task) switched to the CLM objective. The final variant excluded pre-training (w/o Pre-Training) to investigate performance without large-scale unsupervised initialization.

Table 4
Representation approaches and pre-training strategies for all evaluated models.

Model	Representation	Pre-training strategy
TypeMoE	REMI+	SS-MTP
(w/o Type Embedding)	REMI+	SS-MTP
(w/o Multi-Task)	REMI+	CLM
(w/o Pre-Training)	REMI+	–
BERT	REMI+	MLM
TransformerXL	REMI+	CLM
Llama2	REMI+	CLM
BART	REMI+	DAE
MidiBERT	CP	MLM (CP-level)
MusicBERT	Octuple	MLM (Octuple-level; Bar-level)
LakhNES	Note with Time-shift	CLM
MuPT	SMT-ABC	CLM
PianoBART	Octuple	DAE (Octuple-level; Bar-level)

Among the baseline models, MidiBERT relied on the CP representation and applied MLM at the CP level. MusicBERT used Octuple encoding and performed MLM at both the Octuple and Bar levels. LakhNES employed Note with Time-shift under a CLM objective, and MuPT encoded data with SMT-ABC while also using CLM. PianoBART combined Octuple encoding with a DAE objective at both the Octuple and Bar levels. Moreover, BERT, TransformerXL, Llama2, and BART all used the REMI+ representation to examine differences in model architecture under a consistent encoding scheme. BERT was trained with MLM, while TransformerXL and Llama2 used CLM, and BART employed DAE. This setup helped isolate the effect of each architecture on music-related tasks.

As shown in Table 5, the key hyperparameters for each evaluated model included the number of active parameters (Act. Params), hidden size (Hid. Size), intermediate size (Int. Size), number of layers (Layers), number of attention heads (Heads), number of key-value heads (KV Heads), and number of experts (Experts). These values were kept consistent for most models (approximately 0.2B active parameters) so that any performance differences would stem mainly from architectural and pre-training distinctions.

Table 5
Hyperparameters of the evaluated models, including active parameters (Act. Params), hidden size (Hid. Size), intermediate size (Int. Size), number of layers (Layers), number of attention heads (Heads), number of key-value heads (KV Heads), and number of experts (Experts).

Model	Act. Params	Hid. Size	Int. Size	Layers	Heads	KV Heads	Experts
MusicBERT, MidiBERT, BERT	0.2B	1024	4096	16	16	N/A	N/A
LakhNES, TransformerXL	0.2B	1024	4096	16	16	N/A	N/A
MuPT, Llama2	0.2B	1024	4096	16	16	4	N/A
PianoBART, BART	0.2B	1024	4096	8, 8	16	N/A	N/A
TypeMoE (and ablations)	0.2B	1024	2752	12	16	4	4

4.4. Metrics

This section outlines the evaluation metrics tailored to the characteristics of each downstream task. For understanding tasks such as genre and emotion classification, where data exhibited class imbalance, weighted precision, recall, and F1 score were employed to address uneven class distributions. Specifically, accuracy measured the overall correctness of predictions, precision quantified the proportion of predicted positives that were genuinely positive, recall captured the proportion of actual positives correctly identified, and the F1 score provided the harmonic mean of precision and recall. Under this weighting scheme, weighted recall was equivalent to accuracy. Additionally, AUC measured the overall separability between classes, while the confusion matrix provided insights into prediction distributions, facilitating detailed error analysis.

For generation tasks, including continue and conditional generation, Hits@k (with $k \in \{1, 3, 5, 10\}$) evaluated how frequently the correct token appeared within the top k predictions. BLEU, originally developed for machine translation, measured the n-gram overlap between generated outputs and reference sequences, typically computed with $n \in \{2, 4, 8\}$. When different representations were used, Hits@k and BLEU scores were not directly comparable due to differences in tokenization schemes and vocabularies. To enable comparisons among models using different representations, CLAP (Wu et al., 2023) was employed to measure how closely the generated content aligns with a given music fragment or prompt in the embedding space.

4.5. Experimental setup and training details

All experiments involving TypeMoE, its ablation variants, and baseline models were conducted under identical settings. The hardware environment comprised four NVIDIA RTX 3090 GPUs, and the implementation utilized PyTorch in conjunction with DeepSpeed for distributed training and optimization.

Pre-training was performed using the training split of the MidiCaps dataset, consisting of approximately 150,000 samples. A batch size of 16 was employed, and training continued for roughly 60 h, spanning a total of 100,000 steps. Subsequently, each of the four downstream tasks underwent individual fine-tuning as detailed below:

- **Genre Classification:** A random 10 % subset was sampled from both the MidiCaps training and testing splits, resulting in approximately 15,000 training samples and 2,000 testing samples. Fine-tuning was conducted for 10 epochs.
- **Emotion Classification:** The EMOPIA dataset was partitioned into training and testing sets with a 7:3 ratio, yielding around 700 samples for training and 300 for testing. The model underwent fine-tuning for 10 epochs.
- **Continue Generation:** The POP1K7 and POP909 datasets were combined and subsequently divided into training and testing sets with a 7:3 ratio, resulting in approximately 1,800 training samples and 800 testing samples. Fine-tuning was performed for 1 epoch.
- **Conditional Generation:** Like genre classification, a random 10 % subset was drawn from the MidiCaps training and testing splits, providing roughly 15,000 training samples and 2,000 testing samples. Fine-tuning was carried out for 1 epoch.

Furthermore, the model employed for encoding prompts in the conditional generation task utilized GPT-2 (Radford et al., 2019). The CLAP model used to compute cosine similarity was proposed and trained by LAION. Its checkpoint (`music_audioset_epoch_15_esc_90.14.pt`) achieved 90.14 % accuracy on the zero-shot ESC50 task and reached 71 % accuracy on the GTZAN dataset.

4.6. Results

This section presents evaluation results for TypeMoE and its ablation variants, along with baseline models including models such as MidiBERT, MusicBERT, LakhNES, MuPT, and PianoBART, as well as models that used the unified REMI+ including BERT, TransformerXL, Llama2, and BART.

4.6.1. Understanding tasks

Table 6 showed how different models performed on the genre and emotion classification tasks. Observations indicated that TypeMoE achieved strong overall results in these tasks. Compared with baseline models that used the REMI+ representation (BERT, TransformerXL, Llama2, and BART), TypeMoE generally demonstrated higher classification performance, although BERT exhibited comparable emotion classification results.

In contrast, MidiBERT, MusicBERT, LakhNES, MuPT, and PianoBART, which applied original representations, produced varied outcomes. MidiBERT did not explicitly differentiate multiple instrumental tracks, causing difficulties in genre classification but showing favorable results in the emotion dataset, which consisted mostly of single-track piano music. MusicBERT, which adopted Octuple encoding, operated at a medium level of performance. LakhNES and MuPT omitted attributes such as velocity but concentrated on melody or structural aspects, sometimes reaching or surpassing the results of certain REMI+ models in specific classification tasks. PianoBART, which employed Octuple encoding and Octuple- and Bar-level DAE strategy, achieved noticeable improvements over BART.

These differences illustrated how data representation, model structure, and pre-training strategy interacted to influence outcomes. When a fine-grained representation such as REMI+ was paired with models that leveraged multi-track and multi-attribute information, classification accuracy tended to improve. However, if the pre-training objective or structure was not well aligned with symbolic music requirements, the challenges of longer sequences and attribute-related noise could weaken performance. In some scenarios, removing certain attributes or using a more compact representation enhanced efficiency or generalization but was less suitable for tasks relying heavily on multi-track or multi-attribute details. Finally, ablation analysis showed that eliminating type embedding, omitting multi-task training, or skipping large-scale pre-training all led to reduced performance. The most significant drop appeared when no pre-training was conducted, highlighting the importance of large-scale data pre-training for symbolic music understanding.

Table 6
Evaluation results of TypeMoE, its ablation variants, and baseline models on genre and emotion classification tasks. Metrics include accuracy (Acc), precision (Prec), recall (Rec), F1 score (F1), and area under the curve (AUC). Higher scores indicate superior performance.

	Genre classification					Emotion classification				
Model	Acc	Prec	Rec	F1	AUC	Acc	Prec	Rec	F1	AUC
TypeMoE	**71.80**	**69.76**	**71.80**	**70.39**	**84.49**	**76.79**	**76.71**	**76.79**	**76.59**	**84.52**
(w/o Type Embedding)	71.50	69.32	71.50	70.12	84.32	75.89	77.08	75.89	76.23	83.93
(w/o Multi-Task)	70.30	68.28	70.30	68.91	83.45	71.43	72.17	71.43	70.85	80.95
(w/o Pre-Training)	70.10	68.16	70.10	68.72	82.90	63.39	63.63	63.39	63.38	75.60
BERT	69.20	67.47	69.20	67.84	83.06	**76.79**	**77.06**	**76.79**	76.46	**84.52**
TransformerXL	68.90	66.72	68.90	67.49	82.89	70.54	70.38	70.54	70.31	80.36
Llama2	70.60	68.52	70.60	69.30	83.83	66.96	67.07	66.96	66.58	77.98
BART	70.00	68.61	70.00	68.42	83.50	63.39	63.66	63.39	63.45	75.60
MidiBERT	40.30	16.24	40.30	23.15	67.16	74.10	74.90	74.10	73.67	82.73
MusicBERT	67.00	64.68	67.00	65.35	81.85	67.85	68.53	67.85	67.12	78.57
LakhNES	70.10	68.42	70.10	68.56	83.55	68.75	68.80	68.75	68.39	79.16
MuPT	71.00	68.89	71.00	69.55	84.05	68.75	69.19	68.75	68.44	79.16
PianoBART	71.40	69.23	71.40	69.95	84.27	70.53	71.10	70.53	70.67	80.35

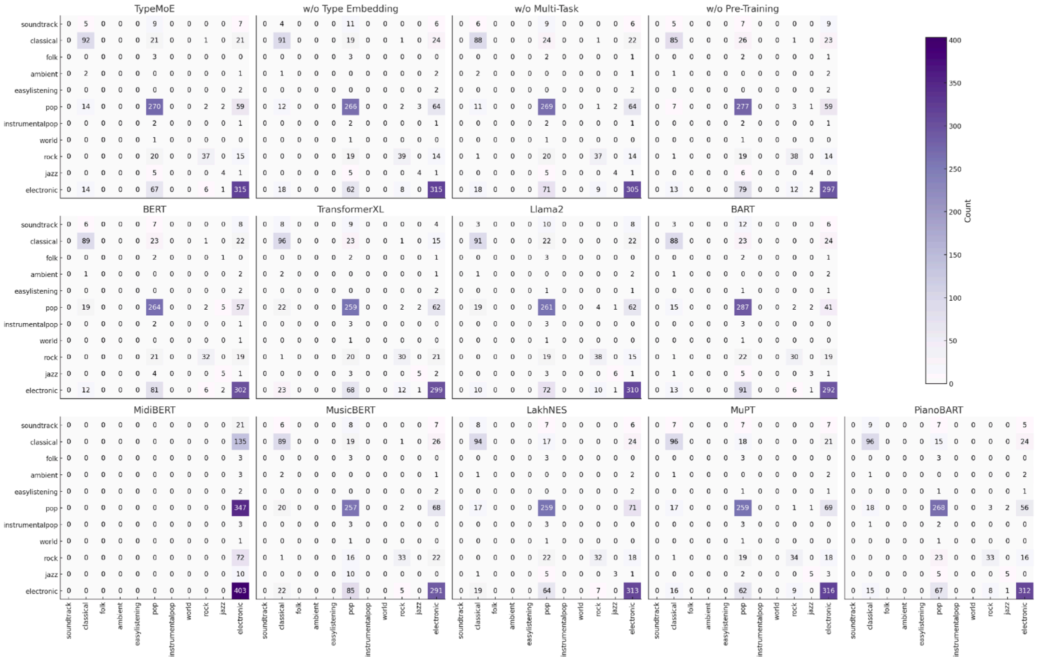


Fig. 4. Confusion matrices demonstrate the performance of evaluated models on the music genre classification task. Each matrix presents predicted labels (horizontal axis) versus ground-truth labels (vertical axis) across 11 genres: *soundtrack*, *classical*, *folk*, *ambient*, *easy listening*, *pop*, *instrumental pop*, *world*, *rock*, *jazz*, and *electronic*. Diagonal entries indicate correct classifications, while off-diagonal entries represent misclassifications.

In addition, Figs. 4 and 5 respectively presented the confusion matrices for genre and emotion classification tasks. Fig. 4 indicated that classical, pop, and electronic represented the largest proportions within the genre classification task, with TypeMoE achieving strong performance in these three genres, resulting in higher overall accuracy. Fig. 5 illustrated that the emotion classification dataset was more balanced, with TypeMoE demonstrating superior performance in Q1, Q2, and Q4. Although BERT attained equal overall accuracy, its performance was notably better in the extreme categories Q1 and Q4 but comparatively weaker in the intermediate category, thereby providing TypeMoE a slight advantage in overall performance.

4.6.2. *Generation tasks*

Table 7 showed that TypeMoE achieved strong results on most metrics in the continue and conditional generation tasks. In particular, it outperformed other REMI+ models on Hits@k as well as BLEU-2 and BLEU-4. However, TransformerXL scored slightly higher than TypeMoE in BLEU-8 and CLAP for the continue generation task. This difference related to the TransformerXL architecture, which used reusable memory in its self-attention mechanism and handled longer sequence spans more effectively. As a result, it had an advantage on BLEU-8 and CLAP. Llama2 showed smaller gaps compared to TypeMoE on both tasks and demonstrated relatively stable performance.

The TypeMoE ablation study indicated that removing type embeddings or multi-task objectives led to lower Hits@k and BLEU scores in continue and conditional generation. Skipping pre-training entirely resulted in a more severe drop, suggesting that large-scale pre-training, type routing, and multi-task learning provided major benefits for symbolic music generation.

For models using their original representation approaches and pre-training strategies, CLAP scores further revealed that LakhNES and MuPT remained competitive in continue generation, and MuPT reached a CLAP score comparable to TypeMoE in conditional generation. Meanwhile, PianoBART, MusicBERT, and MidiBERT showed relatively lower CLAP scores, which were related to their model structures and training objectives. PianoBART employed an encoder-decoder design that

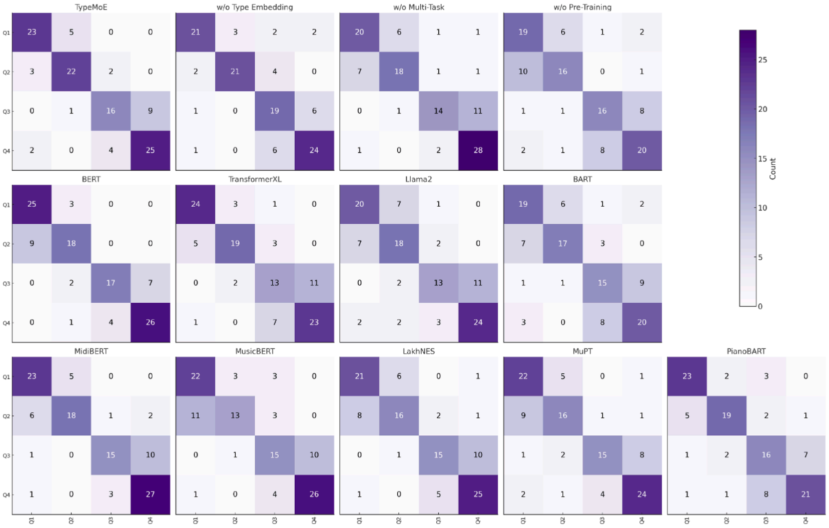


Fig. 5. Confusion matrices illustrate the performance of evaluated models on the emotion classification task. Emotions were categorized according to Russell's Circumplex Model of Affect into four quadrants: Q1 (high valence, high arousal), Q2 (high valence, low arousal), Q3 (low valence, high arousal), and Q4 (low valence, low arousal).

Table 7
Evaluation results of continue generation and conditional generation tasks. Metrics include Hits@k (H@k, with $k \in \{1,3,5,10\}$), BLEU scores (B-n, with $n \in \{2,4,8\}$), and CLAP score. Higher values indicate superior performance.

	Continue generation								Conditional generation							
Model	H@1	H@3	H@5	H@10	B-2	B-4	B-8	CLAP	H@1	H@3	H@5	H@10	B-2	B-4	B-8	CLAP
Ground Truth	–	–	–	–	–	–	–	0.73	–	–	–	–	–	–	–	0.26
TypeMoE	**54.24**	**78.62**	**87.50**	**94.36**	**0.7521**	**0.4999**	0.1848	0.59	**75.79**	**90.04**	**94.31**	**97.90**	**0.8757**	**0.7397**	**0.5265**	**0.24**
(w/o Type Embedding)	51.02	75.59	84.66	92.39	0.7170	0.4469	0.1396	0.54	73.56	88.45	93.12	97.28	0.8620	0.7126	0.4881	**0.24**
(w/o Multi-Task)	52.34	76.90	86.25	93.70	0.7428	0.4884	0.1673	0.58	73.77	89.03	93.74	97.66	0.8647	0.7194	0.4932	**0.24**
(w/o Pre-Training)	22.12	31.83	39.44	52.84	0.0630	0.0041	0.0009	0.13	29.71	46.34	54.29	66.37	0.2326	0.0580	0.0054	0.10
BERT	35.87	56.27	69.15	84.38	0.2131	0.0648	0.0059	0.23	37.62	63.63	75.14	87.98	0.2919	0.1162	0.0160	0.10
TransformerXL	51.99	76.44	85.09	92.93	0.7275	0.4738	**0.1861**	**0.61**	71.08	87.96	93.02	96.67	0.8418	0.6798	0.4405	0.22
Llama2	52.30	77.02	85.90	93.17	0.7471	0.4802	0.1571	0.58	71.49	88.07	93.09	97.38	0.8446	0.6856	0.4450	**0.24**
BART	35.24	61.16	74.74	88.62	0.4184	0.1864	0.0189	0.36	38.78	66.64	79.14	91.81	0.4159	0.2214	0.0520	0.18
MidiBERT	–	–	–	–	–	–	–	0.17	–	–	–	–	–	–	–	0.10
MusicBERT	–	–	–	–	–	–	–	0.23	–	–	–	–	–	–	–	0.10
LakhNES	–	–	–	–	–	–	–	0.51	–	–	–	–	–	–	–	0.22
MuPT	–	–	–	–	–	–	–	0.53	–	–	–	–	–	–	–	**0.24**
PianoBART	–	–	–	–	–	–	–	0.40	–	–	–	–	–	–	–	0.19

focused on local repair rather than full autoregressive generation, making it weaker in global coherence and matching reference music. MusicBERT and MidiBERT both relied on encoder-only setups, emphasizing bidirectional context comprehension but lacking specialized mechanisms for long-range or fully autoregressive music continuity.

Figs. 6 and 7 present examples of music generated by TypeMoE for the continue generation and conditional generation tasks, respectively. In the continue generation task, TypeMoE successfully preserved the pitch range of the original music while effectively maintaining melodic diversity. For the conditional generation task, the generated sample closely matched the specified style, theme, instrumentation, tempo, key, and overall atmosphere provided by the prompt. These results demonstrated that the proposed method not only effectively analyzes the given musical context or textual prompts but also excels in modeling longer-term musical structures.

5. Discussion

This section discussed the implications of balancing multi-task losses and explored potential issues of overfitting and bias associated with dataset usage. The following subsections further elaborate on the practical significance of these findings, propose improvements for future implementations, and outline potential directions for subsequent research.

5.1. Balancing multi-task losses

Fig. 8 presented multi-task loss curves for CLM, KSR, and TSR under different λ settings. The top-left subplot (CLM Loss) indicated that larger λ values slowed convergence, while smaller values ($\lambda = 0$) produced faster convergence. In contrast, the top-right subplot (KSR Loss) and the bottom-left subplot (TSR Loss) exhibited the opposite pattern: larger λ values led to a more rapid decrease in losses, and smaller values caused slower convergence. The bottom-right bar chart summarized validation losses for the three tasks and showed that $\lambda = 1 \times 10^{-2}$ yielded a comparatively balanced performance across CLM, KSR, and TSR. These differences arose mainly because each task exerted a distinct influence on shared model parameters, resulting in varying degrees of coupling and competition. A larger λ placed greater emphasis on meta-information prediction tasks (KSR and TSR), which accelerated their convergence but may have reduced attention to event-level prediction in CLM. A smaller λ reversed the situation by prioritizing CLM at the expense of meta-information guidance.

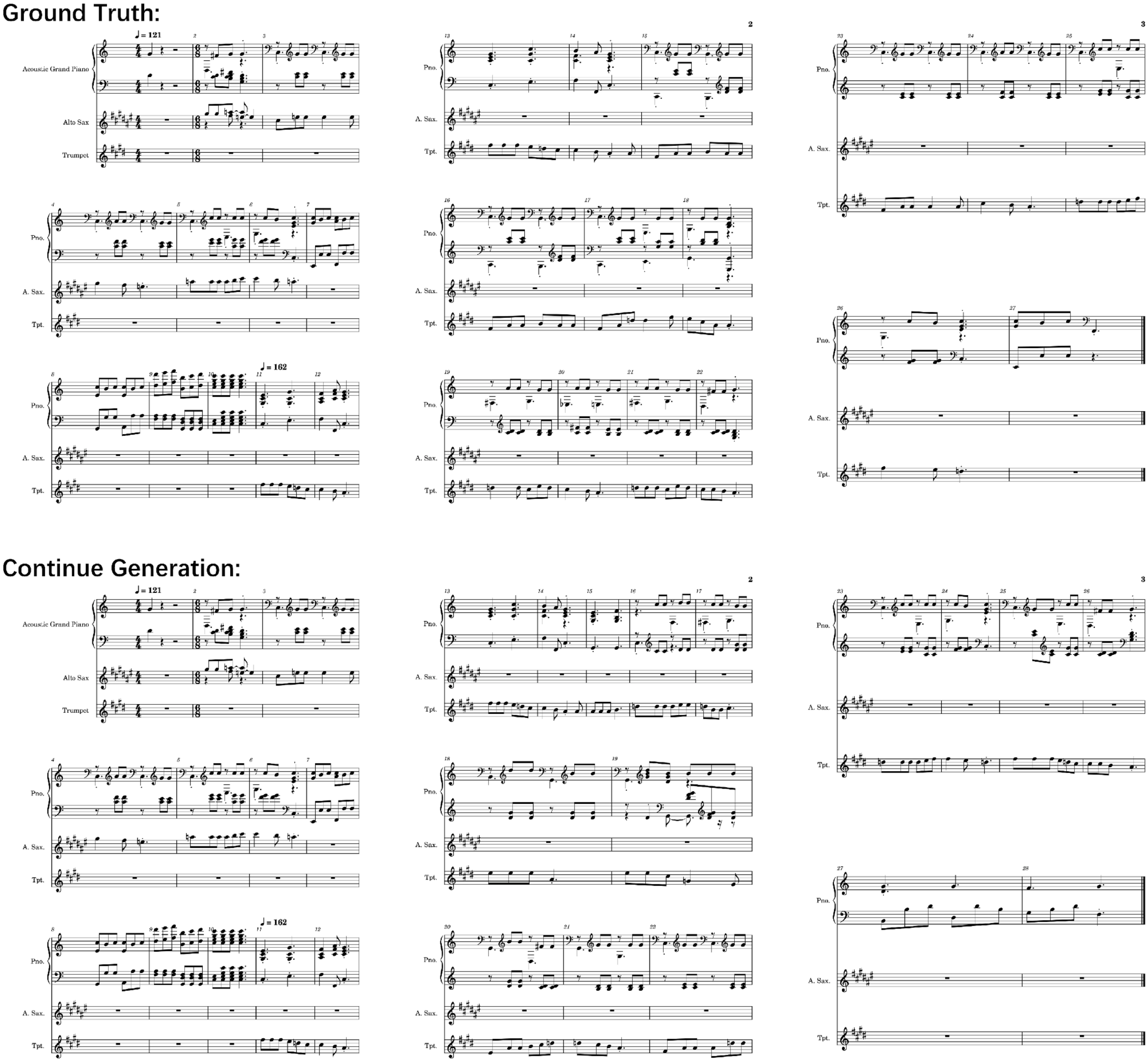
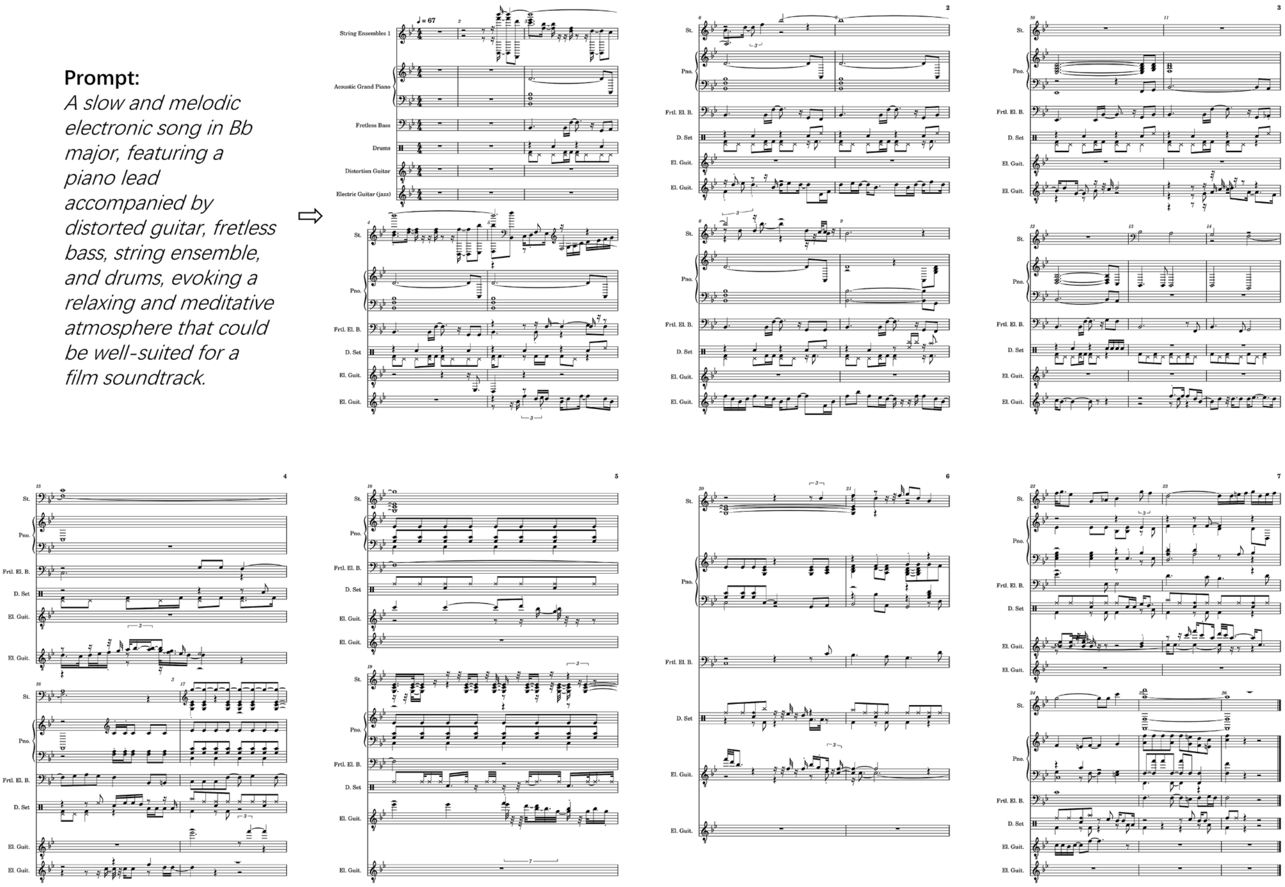


Fig. 6. Illustration of continue generation with corresponding ground truth. The first 14 bars are provided as input context, and the subsequent bars starting from the 15th are generated by TypeMoE.

These observations demonstrated that the multi-task approach was able to incorporate meta-information in a way that enhanced structural understanding while maintaining strong event-level performance, highlighting the advantages of collaboration between tasks. The balanced outcome observed at $\lambda = 1 \times 10^{-2}$ suggested that tuning task weight was necessary under different application requirements and resource constraints. Future research could explore whether larger datasets or additional meta-information (such as form structure or style labels) can further improve the effectiveness of multi-task learning. It might also examine potential negative transfer effects and convergence properties in a broader range of experimental conditions.

5.2. Exploring overfitting and bias in MidiCaps

Although using the same dataset (MidiCaps) for both pre-training and fine-tuning can raise concerns about overfitting and bias, several factors help mitigate these risks. First, MidiCaps is derived from the Lakh MIDI dataset, including 168,385 pieces, and spans a total duration of 8,985 h. This diversity reduces the likelihood of overfitting to narrowly defined patterns. Balanced coverage of various styles and time periods further minimizes potential biases. Second, evaluations on external datasets such as EMOPIA, Pop1K7, and POP909 show that the pre-trained model maintains strong performance, suggesting that it captures broadly applicable musical representations rather than relying on dataset-specific characteristics. MidiCaps also provides distinct training and test splits to avoid overlap between fine-tuning and evaluation. This setup, along with internal validation results, shows no evidence of artificially inflated performance. Moreover, pre-training and fine-tuning serve different objectives, focusing respectively on general representation learning and on downstream tasks such as conditional generation or genre classification. This separation reduces label leakage and overfitting. Future work will consider integrating additional large-scale or cross-domain datasets and applying more rigorous data sampling and deduplication strategies to further enhance generalization and address potential biases.



Prompt:
A slow and melodic electronic song in Bb major, featuring a piano lead accompanied by distorted guitar, fretless bass, string ensemble, and drums, evoking a relaxing and meditative atmosphere that could be well-suited for a film soundtrack.

Fig. 7. Illustration of a conditional generation example.

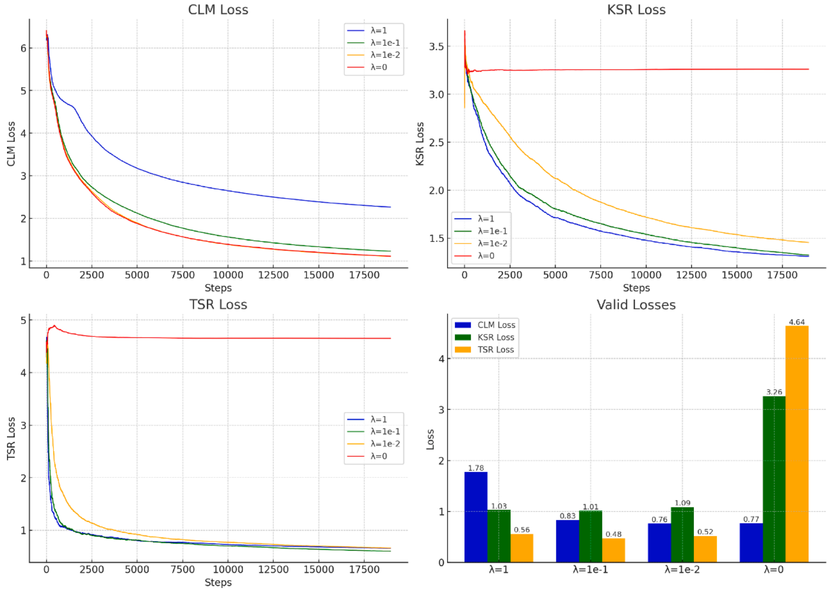


Fig. 8. Analysis of multi-task loss balance during training and validation. The top-left, top-right, and bottom-left subplots illustrated training loss trajectories of CLM, KSR, and TSR tasks, respectively, under varying λ values.

6. Conclusion

TypeMoE and SS-MTP were proposed as a foundational model and pre-training strategy for symbolic music, effectively leveraging type-based routing, multi-task semi-supervised learning, and large-scale pre-training to address both understanding and generation tasks. In genre and emotion classification, TypeMoE surpassed baseline models including BERT, TransformerXL, Llama2, and BART, achieving accuracy and AUC scores of 71.80 % and 84.49 % in genre classification, and 76.79 % and 84.52 % in emotion classification, respectively. Ablation studies demonstrated that removing type embedding, multi-task learning, or pre-training degraded performance, with the absence of pre-training causing the most substantial reduction. In continue and conditional generation tasks, TypeMoE consistently outperformed baseline models, obtaining higher Hits@*k* and BLEU scores. Specifically, TypeMoE reached 54.24 % Hits@1 and a BLEU-2 score of 0.7521 in continue generation, and 75.79 % Hits@1 with a BLEU-2 score of 0.8757 in conditional generation, while also maintaining strong semantic alignment indicated by a CLAP score of 0.24. These results highlighted that capturing type-aware representations, leveraging multi-task learning objectives, and employing extensive pre-training significantly contributed to enhanced symbolic music modeling performance.

Beyond these evaluated tasks, generating coherent long-form compositions remained a significant challenge. Recent advancements in audio-based generative models have demonstrated promising capabilities in synthesizing extended musical pieces, suggesting that addressing long-form generation may be relatively more tractable within the symbolic domain. Potential future directions include investigating advanced gating mechanisms, integrating additional symbolic music tasks such as structured composition, and exploring the incorporation of partial annotations or external domain-specific knowledge to enhance the generalization and expressive capacity of TypeMoE. Furthermore, a notable limitation of this work was the absence of human evaluation, restricting conclusions about perceptual quality and musicality. Future research should thus include listener-based assessments to more accurately gauge model performance from a human-centric perspective.

CRediT authorship contribution statement

Shuyu Li: Conceptualization, Methodology, Software, Validation, Writing - original draft, Writing - review & editing; **Yunsick Sung:** Conceptualization, Methodology, Writing - review & editing, Supervision, Project administration.

Data availability

Data will be made available on request.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) under the Artificial Intelligence Convergence Innovation Human Resources Development (IITP-2025-RS-2023-00254592) grant funded by the Korea government (MSIT).

Supplementary material

Supplementary material associated with this article can be found, in the online version, at 10.1016/j.eswa.2025.128613

References

Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F., Leoni, Almeida, D., Altenschmidt, J., Altman, S., & Anadkat (2023). Shyamal and others, Gpt-4 technical report. Technical Report arXiv preprint. https://doi.org/10.48550/arXiv.2303.08774

Adeoye-Olatunde, O.A., & Olenik, N.L. (2021). Research and scholarly methods: Semi-structured interviews. *Journal of the American College of Clinical Pharmacy*, *4*, 1358–1367.

Ainslie, J., Lee-Thorp, J., Jong, M.D., Zemlyanskiy, Y., Lebrón, F., Sanghai, S., & Gqa (2023). Training generalized multi-query transformer models from multi-head checkpoints. Technical Report arXiv preprint. https://doi.org/10.48550/arXiv.2305.13245

Anthropic (2024). The claude 3 model family: Opus, sonnet, haiku, claude-3 model card.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., & Others (2020). Language models are few-shot learners. *Advances in Neural Information Processing Systems*, *33*, 1877–1901.

Chou, Y.H., Chen, I., Chang, C.J., Ching, J., & Yang, Y.H. (2024). Midibert-piano: Large-scale pre-training for symbolic music classification tasks. *Journal of Creative Music Systems*, *8*(1).

Chowdhery, A., Narang, S., Devlin, J., Bosma, M., Mishra, G., Roberts, A., Barham, P., Chung, Won, H., Sutton, C., Gehrmann, S., & Palm (2023). Scaling language modeling with pathways. *Journal of Machine Learning Research*, *24*, 1–113.

Chu, H., Urtasun, R., & Fidler, S. (2016). Song from PI: A musically plausible network for pop music generation. Technical Report arXiv preprint. https://doi.org/10.48550/arXiv.1611.03477

Dai, Z. (2019). Transformer-XL: Attentive language models beyond a fixed-length context. Technical Report arXiv preprint. https://doi.org/10.48550/arXiv.1901.02860

Devlin, J., Chang, M.W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding (vol. 1). The 2019 conference of the North American chapter. Human Language Technologies.

Donahue, C., Mao, H.H., Yiting, Li, E., Garrison, W., Cottrell, J., & Mcauley (2019). LakhNES: Improving multi-instrumental music generation with cross-domain pre-training. Technical Report arXiv preprint. https://doi.org/10.48550/arXiv.1907.04868

Dong, Y., Jean-Baptiste, Cordonnier, & Loukas, A. (2021). Attention is not all you need: Pure attention loses rank doubly exponentially with depth. In *International conference on machine learning* (pp. 2793–2803).

Ens, J., & Pasquier, P. (2020). MMM: Exploring conditional multi-track music generation with the transformer. Technical Report arXiv preprint. https://doi.org/10.48550/arXiv.2008.06048

Etesami, D., Shirangi, G., Mehrdad, & Zhang, W.J. (2021). A semiempirical model for rate of penetration with application to an offshore gas field. *SPE Drilling & Completion*, *36*, 29–46.

Fedus, W., Zoph, B., & Shazeer, N. (2022). Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, *23*, 1–39.

Hao-Wen, Dong, W.Y., Hsiao, L.C., Yang, Y.H., & Yang, M. (2018). Multi-track sequential generative adversarial networks for symbolic music generation and accompaniment. *Proceedings of the AAAI Conference on Artificial Intelligence*, *32*.

Haviv, A., Ori, R., Press, Ofir, Izsak, P., & Levy, O. (2022). Transformer language models without positional encodings still learn positional information. Technical Report arXiv preprint. https://doi.org/10.48550/arXiv.2008.06048

Hirai, T., & Sawada, S. (2019). Melody2vec: Distributed representations of melodic phrases based on melody segmentation. *Journal of Information Processing*, *27*, 278–286.

Hsiao, W.Y., Liu, J.Y., Yeh, Y.C., & Yang, Y.H. (2021). Compound word transformer: Learning to compose full-song music over dynamic directed hypergraphs. *Proceedings of the AAAI Conference on Artificial Intelligence*, *35*, 178–186.

Huang, C.Z.A., Vaswani, A., Uszkoreit, J., Shazeer, N., Simon, I., Hawthorne, C., Dai, A.M., Hoffman, M.D., Dinculescu, M., Eck, D., & Transformer, M. (2018). Generating music with long-term structure. Technical Report arXiv preprint. https://doi.org/10.48550/arXiv.1809.04281

Huang, Y.S., & Yang, Y.H. (2020). Pop music transformer: Beat-based modeling and generation of expressive pop piano compositions. In *Proceedings of the 28th ACM international conference on multimedia pages* The 28th ACM international conference on multimedia pages (pp. 1180–1188).

Hung, H.T., Ching, J., Doh, S., Kim, N., Nam, J., & Yang, Y.H. (2021). EMOPIA: A multi-modal pop piano dataset for emotion recognition and emotion-based music generation. Technical Report arXiv preprint. https://doi.org/10.48550/arXiv.2108.01374

Jin, C., Wang, T., Li, X., Jie, C., Tie, J., Tie, Y., Liu, S., Yan, M., Li, Y., Wang, J., & Huang, S. (2022). A transformer generative adversarial network for multi-track music generation. *CAAI Transactions on Intelligence Technology*, *7*(3), 369–380.

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., & Zettlemoyer, L. (2019). BART: Denoising sequence-to-sequence pre-training for natural language generation. Technical Report arXiv preprint. https://doi.org/https://doi.org/10.48550/arXiv.1910.13461

Liang, H., Lei, W., Chan, P.Y., Yang, Z., Sun, M., Chua, T.S., & Pirhdy (2020). Learning pitch-, rhythm-, and dynamics-aware embeddings for symbolic music. In *Proceedings of the 28th ACM international conference on multimedia* The 28th ACM international conference on multimedia (pp. 574–582).

Liang, X., Zhao, Z., Zeng, W., He, Y., He, F., Wang, Y., & Gao, C. (2024). Pianobart: Symbolic piano music generation and understanding with large-scale pre-training. In *Proceedings of the 2024 IEEE international conference on multimedia and expo (icme)* The 2024 IEEE international conference on multimedia and expo (ICME) (pp. 1–6).

Liu, X., Ji, K., Fu, Y., Tam, W.L., Du, Z., Yang, Z., & Tang, J. (2021). Technical Report arXiv preprint. https://doi.org/10.48550/arXiv.2110.07602

Madjiheurem, S., Qu, L., & Walder, C. (2016). Chord2vec: Learning musical chord embeddings. In *Proceedings of the constructive machine learning workshop at NIPS 2016* The constructive machine learning workshop at NIPS 2016 Barcelona, Spain.

Mao, H., Shin, T., & Cottrell, G. (2018). Deepj: Style-specific music generation. In *Proceedings of the 2018 IEEE 12th international conference on semantic computing (icsc)* The 2018 IEEE 12th international conference on semantic computing (ICSC) (pp. 377–382).

Melechovsky, J., Roy, A., & Herremans, D. (2024). MidiCaps: A large-scale MIDI dataset with text captions. Technical Report arXiv preprint. https://doi.org/10.48550/arXiv.2406.02255

Modi, S., Lin, Y., Cheng, L., Yang, Liu, G., Zhang, L., & W, J. (2011). A socially inspired framework for human state inference using expert opinion integration. *IEEE/ASME Transactions on Mechatronics*, *16*(5), 874–878.

Qu, X., Bai, Y., Ma, Y., Zhou, Z., Lo, K. M., Liu, J., Yuan, R., Min, L., Liu, X., & Zhang, T. (2024). MUPT: A generative symbolic music pretrained transformer. Technical Report arXiv preprint. https://doi.org/10.48550/arXiv.2404.06393

Radford, A., Wu, J., Child, R., Luan, D., & Amodei, D. (2019). Ilya sutskever, and others, language models are unsupervised multitask learners. *OpenAI Blog*, *1*, 9.

Rae, Borgeaud, J.W., Cai, S., Millican, T., Hoffmann, K., Song, J., Aslanides, F., Henderson, J., Ring, S., Roman, & Young (2021). Susannah and others, scaling language models: Methods, analysis & insights from training gopher. Technical Report arXiv preprint. https://doi.org/10.48550/arXiv.2112.11446

Raffel, C. (2016). Learning-based methods for comparing sequences, with applications to audio-to-midi alignment and matching.

Russell, J.A. (1980). A circumplex model of affect. *Journal of Personality and Social Psychology*, *39*(6), 1161.

Rütte, L.D.V., Biggio, Y., Kilcher, T., & Hofmann, F. (2022). Generating symbolic music with fine-grained artistic control. Technical Report arXiv preprint. https://doi.org/10.48550/arXiv.2201.10936

Saibene, A., Assale, M., & Giltri, M. (2021). Expert systems: Definitions, advantages and issues in medical field applications, expert systems with applications (vol. 177).

Salamon, J., & Gómez, E. (2012). Melody extraction from polyphonic music signals using pitch contour characteristics. *IEEE Transactions on Audio, Speech, and Language Processing*, *20*, 1759–1770.

Su, J., Ahmed, M., Lu, Y., Pan, S., Bo, W., & Liu, Y. (2024). Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, *568*, 127063.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., & Shruti Bhosale, et al., (2023). Llama2: Open foundation and fine-tuned chat models. Technical Report arXiv preprint. https://doi.org/10.48550/arXiv.2307.09288

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., & Polosukhin, I. (2017). Attention is all you need, advances in neural information processing systems (vol. 30).

Walek, B., & Fajmon, P. (2023). A hybrid recommender system for an online store using a fuzzy expert system, expert systems with applications (vol. 212).

Wang, J., Li, S., & Sung, Y. (2023). Deformer: denoising transformer for improved audio music genre classification. *Applied Sciences*, *13*, 12673.

Wang, Z., Chen, K., Jiang, J., Zhang, Y., Xu, M., Dai, S., Gu, X., & Xia, G. (2020). POP909: A pop-song dataset for music arrangement generation Technical Report arXiv preprint. https://doi.org/10.48550/arXiv.2008.07142

Wu, Y., Chen, K., Zhang, T., Hui, Y., Berg-Kirkpatrick, T., & Dubnov, S. (2023). Large-scale contrastive language-audio pretraining with feature fusion and keyword-to-caption augmentation. In *ICASSP 2023-2023 IEEE International conference on acoustics, speech and signal processing (ICASSP)* (pp. 1–5).

Yang, L.C., Chou, S.Y., & Yang, Y.H. (2017). MidiNet: A convolutional generative adversarial network for symbolic-domain music generation. Technical Report arXiv preprint. https://doi.org/10.48550/arXiv.1703.10847

Yu, B., Lu, P., Wang, R., Hu, W., Tan, X., Ye, W., Zhang, S., Qin, T., Liu, T.Y., & Museformer (2022). Transformer with fine- and coarse-grained attention for music generation. *Advances in Neural Information Processing Systems*, *35*, 1376–1388.

Zeng, M., Tan, X., Wang, R., Ju, Z., Qin, T., & Liu, T.Y. (2021). MusicBERT: Symbolic music understanding with large-scale pre-training. Technical Report arXiv preprint. https://doi.org/10.48550/arXiv.2106.05630

Zhang, B., & Sennrich, R. (2019). Root mean square layer normalization. *Advances in Neural Information Processing Systems*, *32*.

Zhang, D., Chen, L., Zhang, S., Xu, H., Zhao, Z., & Yu, K. (2023). Large language models are semi-parametric reinforcement learning agents. *Advances in Neural Information Processing Systems*, *36*, 78227-78239.