Symbolic Music Generation From Graph-Learning-Based Preference Modeling and Textual Queries

Xichu Ma, Yuchen Wang, and Ye Wang, *Member, IEEE*

***Abstract*—This paper investigates the domain of automatic music generation (AMG) and its capacity to produce music that is aligned with user preferences. The incorporation of user music preference (UMP) awareness in AMG technology has the potential to reduce reliance on musicians and domain experts while encouraging users to engage in activities that promote human health and potential. Current research in AMG has been limited to the qualitative control of a constrained set of attributes in the generated music such as selecting a genre from a given list. This constraint makes it challenging to develop music that is both aligned with UMP and suitable for practical text query-based applications. To address this challenge, we propose to apply deep-graph-networks on music community data, jointly modeling UMP and music features. Moreover, users' textual descriptions of expected music can be transformed into graphs that are compatible with UMPs. Node embeddings representing user queries' connotation are extracted to condition the music generator. The results on objective and subjective metrics demonstrate a significant improvement in UMP accuracy by 31.3%, UMP-aware AMG by 63.5%, and text-to-music AMG effectiveness by 76.5%. Our detailed analysis indicates that the generated music aligns best with queries comprised of short sentences and commonly used words.**

***Index Terms*—Music preference, music generation, text to music, graph learning.**

I. Introduction

THE use of appropriate and preferred music as an accompaniment to activities has been observed to encourage individual participation in these activities; this may be beneficial for promoting activities associated with human growth and development. These activities include language learning [1], [2], [3], physical fitness routines [4], and mindfulness meditation [5]. Individuals are likely to engage in these activities with greater efficacy, frequency, duration, and notably, enhanced experiential outcomes [3], [6]. This study seeks to address the three primary obstacles to delivering suitable music for particular activities at any time and place: modeling user music preference (UMP), incorporating UMP into automatic music generation (AMG) systems, and integrating activity-specific requirements as textual queries into AMG.

A common representation of UMP in AMG systems is through a weighted summation of a user's preferred music's features [7]. The UMP embedding is converted into four controlling parameters for a rule-based music generator. The model claims to produce music tailored to users' preferences, but its scope of personalization remains limited as it adheres strictly to widely accepted composition rules for pop music. What is worse, the system lacks the ability to regulate the produced music to meet specific domain-related demands of a task. Queries such as "Provide a fast-paced folk song that is inspiring and preferably sung in choral form," are not feasible because the model cannot connect specific music features with embeddings. The main issue is that current studies oversimplify the analysis of UMP by treating songs as either liked or disliked samples [8], without thoroughly examining the relationship between UMP and the multifaceted features of music.

Another research area aims to create music based on descriptive text. In audio domain, it involves transforming music description into a latent representation and subsequently translating it into musical audio. In audio music domain, Jukebox [9], Mubert [10] and MusicLM [11] project text and its paired audio into a latent space, allowing for music generation through reconstruction and resample from the space. However, their various methods for coupling music and text all exhibit drawbacks such as poor audio quality, computational unscalability, potential copyright infringement, and the need for professional description. In symbolic music domain, current research on symbolic music generation from text is limited to seq-to-seq mapping lyrics or descriptions to melodic notes [12], [13]. However, these attempts fall short in the precise semantic comprehension of descriptions, as well as the effective imposition of expected constraints in music generation.

To address these challenges, we propose to employ deep graph learning to extract UMP by modeling the graph-like relationships between users and their preferred songs. This methodology utilizes the varied feature tags of both users and songs, as well as their internal and mutual relationships. The outcome is a

Manuscript received 26 September 2023; revised 24 February 2024; accepted 25 May 2024. Date of publication 31 May 2024; date of current version 14 November 2024. This work was supported by the Ministry of Education in Singapore under Grant MOE-MOESOL2021-0005. The associate editor coordinating the review of this manuscript and approving it for publication was Professor Richang Hong. *(Corresponding author: Ye Wang.)*

This work involved human subjects or animals in its research. Approval of all ethical and experimental procedures and protocols was granted by the Departmental Ethics Review Committee (DERC), National University of Singapore.

The authors are with the National University of Singapore, Singapore 119077 (e-mail: ma_xichu@nus.edu.sg; yuchen_wang@u.nus.edu; wangye@comp.nus.edu.sg).

Digital Object Identifier 10.1109/TMM.2024.3408060

UMP model that is multi-faceted, interpretable, and compatible with AMG. Additionally, we propose to transform the user's current expectations (in texts) into virtual graphs to refine the UMP embedding. In short, the UMP is built upon the users' music community relationship, and nuanced revisions can be achieved through further descriptions.

Concretely, a music community graph is created by representing users and songs' feature tags as nodes and their relationships (such as liking, being liked, and subscribing) as edges. During the training phase, we collect representative graphs based on the most active users in the community. We aim to enhance a graph-transformer model through self-supervised learning, which involves predicting the presence of relationships between nodes. During the inference stage, a descriptive text query, can be transformed into nodes compatible with the graph constructed for the user. The graph calculates the embedding of the user's node, which includes both the UMP and queried features, as input for a music transformer [14] model. The model generates music that is both UMP-aware and demand-satisfying for human activities.

Our study shows that the proposed graph model improves UMP prediction accuracy by 31.3% compared to the baseline. The utilization of UMP embedding enhances AMG's user likability by 63.5%. Our method enhances the match of expected features between textual queries and generated music by 76.5%. In the comprehensive performance evaluation experiment, our proposed model outperformed various state-of-the-art (SOTA) models in the personalized music generation task, particularly concerning UMP awareness and description correspondence. Furthermore, our thorough analysis of the queries suggests that a concise query comprising commonly utilized words yields music the most aligned with its semantic connotation.

The main contributions of this paper are three-fold:

- We propose to model UMP via multi-facet features of users and songs as well as their relations in the music community. It allows us to examine how preferred songs and subscribed users influence UMPs jointly.
- We present a novel approach utilizing a deep graph learning model and self-supervised optimization to model users, songs, and their relationships. This method extracts UMP embeddings that can be applied to a music-transformer model.
- We are the first to generate personalized symbolic music based on users' textual queries and UMP. This is achieved by converting descriptive texts into graph nodes and updating UMP embeddings to jointly guide AMG.

II. Related Work

A. User Preference Modeling

User preference modeling, which aims to predict and personalize users' experiences, has been widely studied in fields such as music [7], [15], fashion [16], [17], and e-commerce [18], [19], [20] for item recommendation.

Common techniques in this domain include collaborative filtering (CF) [21], [22], content-based filtering (CBF) [23], and hybrid approaches [24] that merge both methods. CF algorithms use user and item information, as well as interaction histories, to connect users with items. This is achieved by either recommending preferred items to similar users or suggesting items similar to the ones users prefer. While effective in recommending satisfying items, this technique is limited in its ability to address tasks beyond recommendation, such as personalized music generation, due to its lack of consideration for the underlying reasons behind user preferences. CBF generates compact embeddings from user interaction data and utilizes them to forecast preferences for new items. Such embeddings encounter challenges in supporting AMG due to the unclear contributing factors in their formation, hindering their effective integration.

Deep learning models, including CNNs [24], [25], RNNs [26], deep belief networks (DBNs) [27], and GANs [28], [29], have been utilized to extract UMPs. Studies have also examined the correlation between UMPs and traits such as emotions and personalities [30]. Due to UMP's long-term stability [31], they are often represented as an average aggregation of preferred artists and songs, or as the primary cluster of samples. In contrast, visual arts preferences, especially in fashion, tend to change periodically due to social trends [16], [17]. Meanwhile, e-commerce preferences change frequently, necessitating the use of reinforcement learning to update preferences in real-time [32], [33]. Research on the integration of UMP is limited to music recommendation. Therefore, it is important to investigate its potential value in more music-generation-related tasks.

B. Music Generation From Texts

Deep learning has gained popularity in AMG tasks, despite being less interpretable and controllable than statistics-based and rule-based methods. CNNs [34], [35], RNNs [36], [37], [38], [39], GANs [40], [41], and Transformers [42], [43] are commonly employed due to their capacity to identify patterns, manage long-term dependencies, and replicate authentic compositions. Language models such as Transformer-XL [44], [45], [46], [47], [48] and GPT-2 [49] have been utilized in AMG to capture longer and structural music dependencies.

Recently, with the advent of ChatGPT[1] and image generation from texts, some works have attempted to use text to condition music generation as well. Although a few attempts of mapping lyrics or descriptions to symbolic musical notes [12], [13] have been made, there is currently no known methods for generating multi-track symbolic music from text. In the domain of text-to-audio, Jukebox attempts to project text and its paired audio into a latent space for reconstruction, enabling new text input to be sampled and decoded into music [9]. This was followed by Mubert [10] which maps user input to a tag list, finding matching audio segments for concatenation. Recently, MusicLM [11] maps long texts to a cross-model audio-text embedding through Mulan [50] and decodes the semantic and musical embeddings together into new pieces.

Despite the progress made in text-to-music generation methods, existing approaches still face several limitations such as: (1) Potential copyright disputes, since these methods involve

[1][Online]. Available: https://openai.com/blog/chatgpt

sampling and concatenating pre-existing audio clips rather than composing a new piece. (2) Low audio quality: they mainly employ a single track with a low sample rate of 24 kHz and a 6 kbps compression, and thus are subject to the substandard quality of outdated audio files and the considerable reliance on computing resources. This leads to suboptimal tone quality, especially for music genres that demand superior audio separation, such as orchestral ensembles. (3) User-friendliness and personalization, since generating satisfactory personalized results necessitates extensive, precise, and comprehensive textual input, which can be a daunting task for the majority of users. Disparities in listener perceptions and language usage when describing music can result in biases and systematic errors.

C. Controllability in AMG

The foundation of a demand-satisfying AMG is controllability. Controllable music generation allows users to customize or control the generated music by specifying certain parameters or characteristics. This method allows for a specific and interactive generation process that cannot be achieved through traditional AMG techniques. The generation should ideally be able to understand and reflect free-form descriptions from users.

A typical working paradigm of control can be found in many controllable music generation systems where various constraints are encoded as conditional embeddings and infused into deep-learning based models to control AMG. Prevalent conditions include emotions [51], styles [52], and text descriptions [53] as well as two widely used techniques include embedding infusion [48] and disentanglement [40], [54].

1) Categories of Control: a) Emotion: Music generation based on emotion labels utilizes valence-arousal dimensions to classify emotions such as happy, calm, agitated, and suspenseful. Emotion-conditioned music generation has been previously explored through deep generative models [55], [56]. These early approaches utilize a generator and an emotion classifier to control polyphonic music generation based on specific emotions. Subsequent efforts have been made to further integrate emotions into generated music. Emotional vectors are commonly used to encode emotion tags and guide the generator in the process [51], [57], [58], [59], [60]. Certain algorithms are designed for particular purposes, such as generating background music for role-playing games [61], producing audio books [62], or providing musical accompaniment for images [59]. However, most existing works have a limited range of emotion tags, which does not meet our need for generating descriptions freely.

Style: Styles can serve as a form of guidance too. Music in works guided by this condition is created either by following fixed target styles [52] or by using styles obtained from existing composers and pieces [54], [63]. Certain musical compositions produce melodies that are dependent on other musical elements, such as chords [34], [64], [65], lyrics [66], [67], [68], or a leading theme [69]. Conversely, these elements can also be dependent on the melody [70], [71], [72], [73]. The existing techniques utilize datasets with interdependent stylistic parameters. However, the underlying logic remains unclear, and these parameters are not labeled for training. Consequently, it is uncertain what the neural network learns for these parameters. This limits controllability and describability. Further, certain studies have demonstrated that users can modify musical attributes, such as pitch contours and rhythmic complexity. However, such modification can only apply to existing pieces [54], [74].

Text: Textual inputs are utilized as conditions to enhance the descriptive capability of music generation [53]. An active learning based music generation model with pre-embedding has been proposed and trained with the Lakh Pianoroll Dataset [40], [75]. The BERT model [76] is utilized to embed input texts, which are subsequently translated into musical output. The model is trained to generate diverse outputs based on texts with varying semantics. The lack of semantic connection between words and music, however, results in an inexplicable and unpredictable text-to-music flow, as there exist arbitrary mappings from different points in the distribution to different generated pieces.

2) Techniques of Control: a) Embedding Infusion: Controllable music generation can be achieved through an embedding infusion technique that utilizes condition embeddings to regulate the music generation process. Two types of embeddings exist: discrete tokens [48], [77], [78] and continuous tokens [57]. Meanwhile, infusion can occur in two ways: concatenation in the input tokens or in the hidden states [57]. We utilize continuous tokens to prevent information loss caused by binning of continuous value or truncation of the generated sequence. Conditioning the model on a specific embedding enhances the ability to control the generated music's characteristics, thereby serving as a potent manipulation of music generation and synthesis [79], [80], [81], [82].

Disentanglement: Disentanglement pertains to the partitioning and grouping of distinct musical features in controllable AMG, enabling users to designate and regulate particular facets of the produced music. Considerable research has been dedicated to the creation and assessment of disentangled models for music generation. These models aim to separate musical elements and characteristics, such as melody, harmony, and rhythm [54], [74]. However, these works typically concentrate on a fixed set of factors and are difficult to accommodate unrestricted description.

To conclude, current methods are limited in their capacity to generate personalized and high-quality music. (1) Current UMP models face challenges accommodating multi-facet user preferences of music characteristics, leading to a mixed-up generation result of all listened music. (2) Existing text-to-music generation is inadequate in generating high-quality symbolic music that allows secondary editing or collaboration between humans and computers. (3) The imposed conditions impact only a fraction of music characteristics.

III. Methodology

In this study, we present a novel method for generating music that is tailored to individual tastes and has characteristics that are in line with textual queries. The three modules of our methodology are UMP extraction, query conversion, and music generation. Fig. 1 displays the system's workflow. First, we propose a graph-learning based model for UMP extraction, which

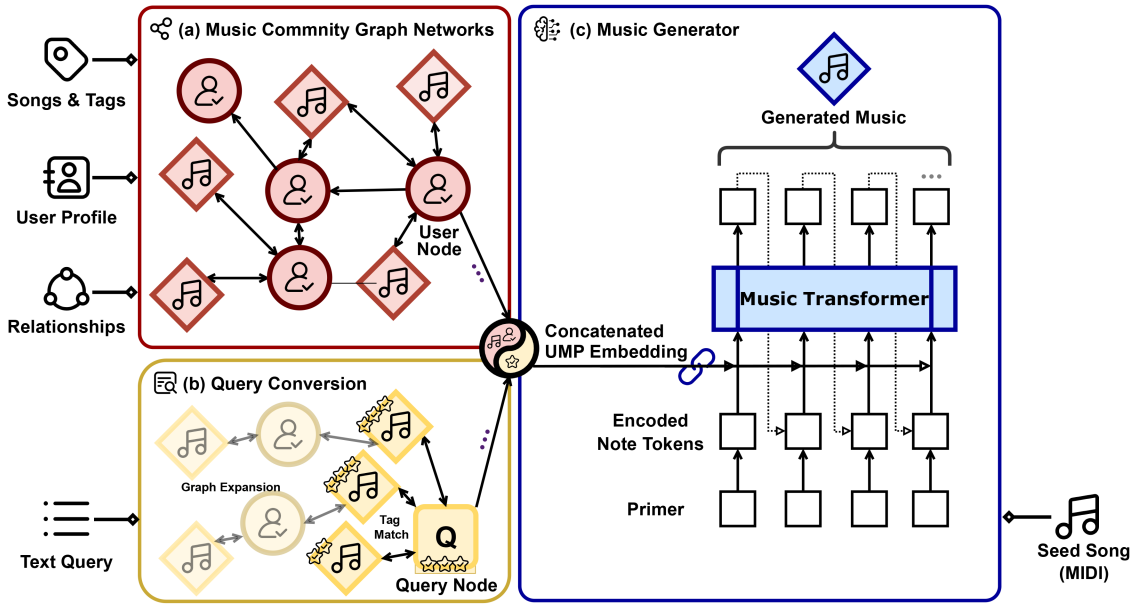

Fig. 1. System framework. (a) Music sharing community graph. The graph comprises user and song nodes. The edges are directional and can be classified into three types: user-user, user-song, and song-user. (b) Textual query-converted graph. The process of converting a textual query into a graph involves creating a center query node, expanding the graph by connecting relevant nodes to the query node, and executing the graph propogation. The center node's embedding is desirable to condition music generation. (c) Multi-facet UMP conditioned music generator.

represents users and songs in a music sharing community as two types of nodes and describes their connections using three types of directed edges Fig. 1(a). We suggest using unsupervised optimization to enhance the ability of the extracted UMP to forecast specific users' preferences for particular songs or other users. This architecture resolves the challenge of organizing multi-type data in UMP models by utilizing a graph that inherently depicts users, songs, and their topological relations. The foundation of a UMP is a specific user's node embedding. By creating new nodes from the user's textual query and connecting relevant song nodes to the user node Fig. 1(b), we can further grow the graph based on the fundamental UMP. After information propogation within the graph, the user node displays the optimized UMP features with the queried attributes. To generate a multi-track musical composition based on the embedding of the user node Fig. 1(c), we propose a transformer-based music decoder. Since it complies with the domain-specific requirements outlined in the input text and is also in line with the UMP embedding, the generated music is expected to encourage participation in beneficial activities.

A. Problem Formulation

The process of generating UMP-aware music from text can be divided into three formal stages:

UMP Extraction: A graph $G\{V, E\}$ is constructed to represent a music community, given a set of users $U = \{u_1, u_2, \ldots, u_N\}$, a set of music pieces $M = \{m_1, m_2, \ldots, m_K\}$, and their likability relations. The sets V and E represent the nodes and edges respectively, where $|V| = |U| + |M|$. User nodes are initialized with their profiles, and song nodes are initialized with the bag-of-word encoding of their tags. The UMP extraction model $O_\theta(G)$ generates embeddings $h \in \mathbb{R}^d$ for all nodes, representing the users' fundamental UMP and the songs' properties. The trainable parameters of the UMP extraction model are denoted by θ.

Query Conversion: The provided textual query, denoted as $q = \{t1, t2, \ldots, t_F\}$, wherein each t_f signifies a lexical token, delineates the user's expectation of the music to be generated. This query undergoes a process of refinement into a set of tags, subsequently conversion into a virtual node that encapsulates the position and context of such expected songs that match the textual query in the music community. During the conversion process, the pertinent song nodes are incorporated into the graph denoted as G and subsequently connected to either a virtual query node or an existing user node, contingent upon the user setting. This conversion yields the graph G'. To obtain an embedding that encapsulates the queried features and the user's UMP, the virtual/user node's embedding h_u is updated to $h_u{}'$ through a recalculation of graph propagation in the updated graph $O_\theta(G')$.

Personalized Music Generation: Conditioned on the the embedding of a node h_u and optionally seed music as primer p, the music generator $A_\gamma(h_u, p)$ composes a new music piece m' that conforms to the UMP and satisfies the desired characteristics outlined in the query. γ is the music generator's trainable parameters.

B. Graph-Based UMP Embedding Model

We propose a graph transformer-based UMP embedding model for learning song embeddings and users' musical preference embeddings. The model is designed to jointly learn both user profiles and musical features of songs from community

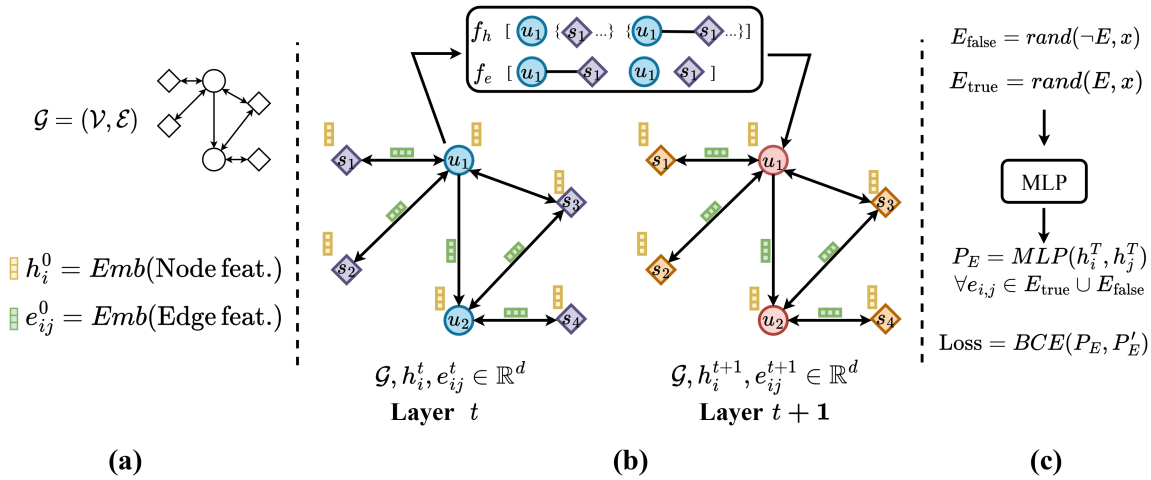

Fig. 2. Graph network architecture. (a) Graph construction. (b) Graph model feedforward propagation of one layer. (c) Graph model optimization.

data. The optimized embeddings can serve as conditions in UMP-aware music generation.

1) Graph Creation: This study involves the construction of a graph to depict the entities and their relationships within a music community. The graph comprises user and song nodes. The edges are directional and can be classified into three types: user-user, user-song, and song-user. For instance, when user a follows user b, a directional edge is created from b to a to account for the influence of b on the computation of a's embedding during graph propagation. Likewise, when a user likes a song, it results in the formation of mutual connections between them.

Song and user nodes possess associated features utilized in the calculation of the embeddings. Song features include tag list, play count, song name, artist name, and album name. Meanwhile, user features include the username profile and the country. All features are projected into a hidden space and represented as vectors of equivalent dimensions for graph propagation [83].

The graphs constructed for training typically revolve around "center users" who serve as the initial nodes in a community network. According to our findings of the datasets, followers of active users also frequently engage in active behavior. Ideal "center users" have a ton of followers, tags, and favorite songs, which creates more graph edges and minimizes the chance of underfitting. The most active, tag-diverse, and impactful individuals are selected as the "center users". Next, graphs are expanded with k hops by connecting more preferred songs and followed users of the peripheral nodes. In inference, an exclusive graph of a user is built similarly starting from this newcomer user as the center.

2) Model Architecture: Our UMP embedding model is based on the graph transformer proposed in [83]. The graph input consists of two node types (users and songs) and three edge types (user-to-user, user-to-song, and song-to-user). The nodes and edges' embeddings are vectors of dimension d, collecting information from neighboring nodes and iteratively updating for T rounds (layers). As shown in Fig. 2, each round of graph propagation is parallelly conducted to nodes and edges. Afterwards, we take the final embeddings of user nodes as their fundamental UMPs.

Node and edge embeddings (h and e) are initialized as stacked feature embeddings that undergo linear mappings:

$$h_{m_i}^0 = L(concat(emb(\text{tags}), emb(\text{play count}), emb(\text{song name}), emb(\text{artist name}), emb(\text{album name}))) \quad (1)$$

$$h_{u_i}^0 = L(concat(emb(\text{username}), emb(\text{country}))) \quad (2)$$

$$e^0 = L(emb(\text{edge type})) \quad (3)$$

where emb represents a linear mapping from the feature's original size to the hidden dimension, L represents another linear mapping from the concatenated feature size to the hidden dimension d, and edge type is in the set 0,1,2 denoting the three edge types.

Then the attention in the graph of each round is formulated as:

$$\begin{aligned}&\text{Attention}\left(h_i^t, h_j^t, e_{i,j}^t\right)\\&= \text{LeakyReLU}\left(\mathbf{A}^T\left[\mathbf{W}h_i^t || \mathbf{W}h_j^t || \mathbf{W}e_{i,j}^t\right]\right)\end{aligned} \quad (4)$$

where h_i^t and h_j^t are the hidden states of nodes i and j, $e_{i,j}^t$ is the edge embedding of the edge i $\rightarrow$ j at round t, $\mathbf{W}$ is the learnable weight matrix, $||$ denotes concatenation, and $\mathbf{A}$ is a learnable weight vector. Next, the attention scores between nodes are computed as:

$$\alpha_{i,j}^t = \frac{\exp\left(\text{Attention}\left(h_i^t, h_j^t, e_{i,j}^t\right)\right)}{\sum_{k\in\mathcal{N}(i)}\exp\left(\text{Attention}\left(h_i^t, h_k^t, e_{i,k}^t\right)\right)} \quad (5)$$

where $\alpha_{i,j}^t$ is the attention score between node i and j, and $\mathcal{N}(i)$ is the set of nodes that are directed towards node i.

Last, the embeddings are updated with the attention scores:

$$h_i^{t+1} = \sigma\left(\sum j \in \mathcal{N}(i)\alpha_{i,j}W_h h_j^t\right) \quad (6)$$

$$e_{i,j}^{t+1} = \sigma\left(\sum k \in \mathcal{N}(i)\alpha_{i,k}W_e e_{i,k}^t\right) \quad (7)$$

where h_i^{t+1} and $e_{i,j}^{t+1}$ are the updated hidden states of node i and edge (i, j), W_h and W_e are learnable weight matrices

and σ is the tanh function. After the graph propagation of T rounds, h vectors of user nodes contain the information collected and aggregated from the users' favorite songs and the preferences of other users they follow, thus representing their own UMPs.

3) Self-Supervised Training Procedure: We utilize self-supervised learning to train the UMP embedding model for predicting user-song connections without labeled user preference data. The edge prediction task is employed to optimize the model with the edge data created from music communities. The model is given an input graph and randomly selects x existing and x non-existing edges as positive and negative samples (denoted as E_{true} and E_{false}). The model then predicts the presence or absence of these edges based on the embeddings of their starting and ending nodes. With one graph fed into the model per batch, we implement a random sampling strategy for the $2x$ user-song edges to mitigate overfitting risk.

$$\begin{aligned} E_{true} &= rand(E, x),\ E_{false} = rand(\neg E, x) \\ P_E &= MLP(h_i^T, h_j^T),\ \forall e_{i,j} \in E_{true} \cup E_{false} \\ Loss &= BCE(P_E, P'_E) \end{aligned} \tag{8}$$

where P'_E denotes the ground truth of the sampled edges' existence.

C. UMP-Aware Music Generation Model

We design a UMP-aware music generator that incorporates UMPs and primers (seeds) to generate personalised music. The music generator utilizes a user's UMP embedding and an optional primer as leading notes to produce music. UMPs are obtained from the acquired node embedding h. While our model can generate music from scratch, we consider primers to be a supplementary sample representation of the UMPs, despite their non-necessity.

1) Music Encoding: To obtain a structured representation of symbolic music, we utilize the performance encoding proposed by [84] which includes 128 NOTE-ON events, 128 NOTE-OFF events, 100 TIME_SHIFTs allowing for expressive timing at 10 ms and 32 VELOCITY bins for expressive dynamics. The encoding is implemented through a neural processor[2] which converts MIDI files into numerical representations of musical events, extracting information such as timing, velocity, pitch, and instrument, as shown in Fig. 3. The resulting arrays provide a compact sequential representation of music and can be combined with UMP embeddings for further processing.

2) Model Architecture: The music generation model utilizes a MusicTransformer architecture, inspired by the work of [14], to produce music that resembles human-composed music in both structure and aesthetics as well as having a long duration. We select it because it is the state-of-the-art symbolic generation model. Its autoregressive music generation method also aligns well with our idea of incorporating UMP for precise and multi-faceted control.

[2][Online]. Available: https://github.com/jason9693/midi-neural-processor

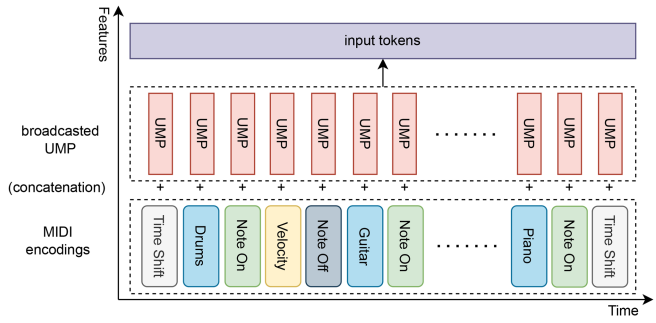

Fig. 3. Data encoding. The input MIDI is encoded into a sequence of musical events where each token is concatenated with identical UMP embedding.

3) Training Procedure: The model was pre-trained on the LPD-5-full dataset as presented by [40], [75]. The dataset comprises samples with a five-track structure (Piano, Drum, Strings, Bass, and Guitar), facilitating the generation of multi-track music with long-term structures. After getting familiar with the dataset's general music style, the music generator is fine-tuned using a subset data of 470 songs and their corresponding node embeddings, which represent the listeners' UMPs. The songs' node embeddings are integrated into the input music token sequences prior to being fed into the music transformer, as depicted in Fig. 1(c). The embedding is appended to each time stamp of the input sequence to minimise information loss, as discussed in Section II-C-2.

D. Music Generation From Textual Query

Our UMP embedding model enhances controllability over AMG by leveraging the graph model's capacity to integrate diverse data types. AMG can be conditioned by the embedding of a song node, a user node, or a combination of both. Various methods for graph construction and embedding computation can direct AMG to generate music in accordance with different requirements. For instance, to produce a pop song that aligns with a user's preference, we limit the first hop of the user's graph to pop song nodes exclusively.

Moving forward, to generate music that aligns with the desired characteristics described in the query, we propose to model the textual queries as narrowed-down forms of UMPs. Textual queries can be converted into graphs to obtain their equipotent UMP embeddings. The query-converted UMP embeddings can control music generation to reflect the desired characteristics of the query. The process involves creating a center query node, expanding the graph by connecting relevant nodes to the center node, executing the graph propogation, and using the center node's embedding to conditionally generate a song.

To obtain the UMP embedding of a textual query, initially, we analyze the query text and rank the tags based on their semantic relevance to the query using BERT [76]. Specifically, extracting attributes from textual queries involves establishing a semantic context match between a song's tag repository (a 1×128 vector $\in \{0, 1\}$) and the query text. To accomplish this, we leverage the BERT model to compute word embeddings for both the tags in the song tag repository and the user query. These embeddings

enable a sorting mechanism that ranks the tags based on their relevance to the query:

$$\arg\max_i [BERT(t_i) \cdot BERT(query)] \tag{9}$$

Subsequently, a center query node is created. It can be initiated as a virtual graph node that represents the query or copied from an existing user node. Then we select the top K tags with the highest relevance ranks, and thereby activate the most semantically relevant tags of the virtual/user nodes' initial feature vector.

Next, the graph for this query is further expanded from the center virtual/user node. Specifically, we filter out the songs nodes that have the highest cover ratio of tags with the top K tags from the dataset, and connect them to the center node. Then the graph is expanded in the same way as described in Section III-B1. Finally, the embedding of the center node is derived from the graph transformer's propagation computation and utilized to condition the music generation.

Advantageously, a query node embedding is compatible with the embeddings of both users and song nodes' UMP embeddings. The query can either condition AMG alone by connecting to virtual center node or function as a supplement to a user's fundamental UMP by connecting to an actual user node. This reduces the precision demands for the description in text-to-music generation. For example, if a user inputs the query "A fast-paced classical song", and provides some songs as references, but the songs that the user provides as "classical" are actually contemporary symphonic and those provided as "fast" are actually what other users consider as medium speed, the use of the graph network ensures that for this compound embedding, medium-speed contemporary symphonic songs are tagged as "fast" and "classical", so the user's requirements are embedded correctly from the user's subjective point of view.

Our method improves upon existing text-based music generation by utilizing the user's network as the fundamental control, thus obviating the requirement for musical proficiency. Our model incorporates both textual inputs and UMPs to mitigate biases arising from imprecise music descriptions and the musical perceptions that vary from person to person. This achieves a more personalised generation process for individual users. The utilization of a symbolic music generator enables genuine composition and amplifies the possibility for subsequent editing and collaboration between humans and machines.

IV. Experiment Configuration

A. Implementation Details

Our UMP embedding model is implemented based on the graph transformer described in [83]. The hyperparameters, such as the number of layers T of 10, the number of attention heads of 8, and the utilization of residual networks, remain unchanged. The tag number, denoted as K, is set to 3 for textual-query graph construction. Given that most songs in the Last.fm dataset possess between 3 to 5 tags, we assert that this lower bound selection optimizes representativeness and compatibility. Consequently, this approach mitigates the issue of certain perfect match neighbors being erroneously neglected due to their lower tag count thus lower tag cover ratio. The hidden dimension d is set to 64, which is equivalent to the output embedding dimension. Binary cross-entropy loss is utilized for edge prediction. An Adam optimizer with βs=(0.9, 0.999) is utilized to update the parameters, with an initial learning rate of 7e-4.

We implement our music generation model with the Music-Transformer [14]. The model adheres to the configuration of 6 layers, 8 heads, 512 hidden dimensions, and 1024 feedforward dimensions. The UMP embedding is integrated into the input music sequence through the conditioning method of concatenation. Cross-entropy loss is utilized to evaluate the dissimilarity between the generated sequence and the ground truth. The model with the highest validation accuracy is selected for practical inference.

B. Data Preparation

1) Data Collection: We collected 3,477,790 songs and 79,914 users from Last.fm,[3] a music platform that enables users to monitor their listening histories and subscribed users, as well as explore new music. The platform offers APIs to access pertinent data on songs and users. To ensure diversity and representation in the dataset, we initially selected 50 common songs that correspond to the top tracks of the 50 most popular tags. The commenters who give the highest ratings on these songs are referred to as "initial users." We expanded "music networks" of our "initial users" by utilizing Last.fm's API to gather information on their followers and followees, and identified the most active users within these networks with the most listening events. These users are defined as "centre users". Our collected data comprises the graphs expanded from "center users" with a radius k of 5 hops.

2) Feature Selection: User and song nodes' features gathered from Last.fm involve users' name and country, and songs' name, album, artist, play count, and tags (127 most popular tags are taken into account). The 127 tags are represented as bag-of-word vectors of 127 dimensions. Future work may expand the available information beyond the current limitations of the last.fm API. The model's robustness to changes in graph features enables easy updating and retraining without altering the model architecture.

3) Graph Division: To optimize the efficiency of training, we divided large graphs into subgraphs containing approximately 10k nodes each. To maintain the potential connections between subgraphs, we adopted a 50% overlap rate of the songs and users included in the subgraphs. This training strategy allows us to train the model on smaller, more manageable subgraphs while still capturing the complexity and connectivity of the larger graphs [85].

V. Subjective Experiment

The purpose of this experiment is to determine whether the suggested UMP embedding model and the suggested music production model are effective in modeling UMPs and producing music that fits those UMPs. We also seek to test how well our

[3][Online]. Available: https://www.last.fm/

algorithm can produce music based on user-provided textual queries. Building upon this foundation, we aim to assess the overall performance of the proposed model in personalized music generation tasks that require simultaneous consideration of UMP awareness and textual descriptions.

A. Evaluation of UMP

1) Compared UMP Modelings: We conducted a comparative analysis of various UMP acquisition methods to determine the optimal approach for constructing a graph that produces accurate UMP embeddings. These methods for extracting UMP embedding differ in their focus, including feature tags, preferred songs, subscribed users, and listening histories, including:

- $U2A$: based on preferred tags—the user node in the graph connecting to the song nodes that match the tags the most;
- $U2B$: based on preferred songs—the user node in the graph connecting to the preferred song nodes;
- $U2C$: based on subscribed users—the user node in the graph connecting to the users he/she follows;
- $U2D$: the average of ($U2A$, $U2B$, and $U2C$)
- U_{rand}: a randomly generated vector as the control group
- $U1$: derived from the state-of-the-art UMP model M_1 proposed in [7]. It calculates UMP embeddings as the weighted summation of users' listening histories.

2) Measurement of UMP Embeddings' Effectiveness: To assess the compared UMP models' performance, a downstream task is chosen to predict the likelihoods of user-song edge's existence based on their node embeddings. The hypothesis posits that the stronger the predicted likeability of a user-song edge, the more likely the user likes the song. To conduct the test, we chose the most recommended songs associated with each UMP that result in the highest edge likability, and selected the least recommended songs accordingly. The UMP embedding model's effectiveness is evaluated by measuring users' ratings on the songs. Specifically, we propose incorporating an offset likability factor in the selection of top recommended songs to mitigate the influence of highly popular songs:

$$\text{likability}_{\text{offset}} = \text{likability} - \text{likability}_{\text{base}} \tag{10}$$

where $\text{likability}_{\text{base}}$ is the likability between songs and a zero-initialized vector, representing the popularity of songs under mainstream music preference. Subtracting $\text{likability}_{\text{base}}$ from predicted likability minimizes the dominating influence of popular songs and facilitates the identification of users' individual preferences.

Higher ratings obtained by top recommended songs (those with the highest predicted offset likability values), lower ratings obtained by the least recommended songs (those with the lowest predicted offset likability values), and the large net difference between the high and low ratings all indicate better performance of our proposed UMP model. They all show that the model grasp the UMP and can predict songs they would like/dislike.

B. Evaluation of UMP-Aware AMG

Our attention is also directed towards examining the effectiveness of integrating UMP into AMG. We conducted tests on different combinations of UMP models and primers (leading music) that occur in various AMG scenarios, such as generating music for non-professional individuals and collaborating with musicians using their input primers.

We have selected the following primers for comparison:

1) $P1$, a song preferred by the user from the LPD-5 dataset
2) $P2$, a randomly selected song from the LPD-5 dataset
3) $P3$, no primer, music generation from scratch

Measurement of UMP satisfaction. Each compared UMP embedding is used as a condition to generate three songs from the three primers. There are two candidate methods of UMP embedding conditioning music encodings. The "adding" operation combines the tensors of UMP embedding and music encoding element-wise by summing their corresponding elements while the "concatenation" operation appends them along a specified axis, creating a new tensor with double dimensions.

A preliminary case study was conducted to compare the generated music conditioned by "adding" or "concatenation" operations. We have opted to adopt "concatenation" in our system and experiments, as it produces musically more satisfactory music (rating 3.2) compared to the "adding" (rating 2.5) operations, based on the feedback received (+28%). The addition operation is appropriate when considering one input as a residual "correction" or "delta" to the other input. Concatenation, on the other hand, is preferable when the two inputs present less element-wise relation. We argue that UMP affects music encodings holistically rather than on an element-wise basis, elucidating the advantage of concatenation in our experiments.

A total of 18 generated songs are listened to and rated by users on a ten-point scale to determine the optimal UMP-primer combination in different AMG scenarios. We analyze the ratings to determine the combinations of UMPs and primers that are most effective for vast audiences.

C. Evaluation of AMG From Textual Queries

Our evaluation involves assessing our system's ability to produce music based on textual queries. Besides the six UMPs above, we have developed a UMP embedding $U2T$, which utilizes tags converted from a textual query. It is derived from a graph where the user node connects to the songs that are the closest to the converted tags. Three songs are generated from $U2T$ using three primers for comparison.

Measurement of Textual Query Satisfaction. U_{rand} serves as the control group for assessing the effectiveness of $U2T$ in comprehending textual queries and producing appropriate songs. Users will rate songs produced by two UMPs based on description correspondence to a given query. Higher ratings are anticipated for the songs generated by $U2T$ if it can effectively capture meaningful semantic associations from the query.

D. Evaluation of Personalized AMG

For a comprehensive evaluation of our proposed optimal personalized music generator (U2D+P_2), we opted for representative models in the same category as baseline models. These include the text-to-symbolic-music generator based on a language model BART [12] (recognized as superior among a series

of language models in music generation tasks [12]), a commercial symbolic music studio WavTool[4] based on GPT-4, and a rule-based UMP-aware music generator [7]. These three models either represent the SOTA in text-to-symbolic-music generation or UMP-aware music generation.

In this experiment, participants provide a textual description of the desired song, which is then input into the comparative BART model [12] and WavTool for music generation. The BART model generates ABC-notation format songs from text, which are rendered into audio. In WavTool, we consistently pick the primary recommendation generated by the system based on the description, resulting in lead sheet music comprising melody, chord, and bass components. Subsequently, following the UMP embedding extraction of U2D in subsubsection V-A-1 and the equiponent UMP embedding calculation of textual queries described in subsection III-D, a concatenated UMP embedding of U2D and the input query is fed into our adopted music generator (MusicTransformer-based [14]) for music generation. Finally, the participants' extracted UMP embedding U1 is input into [7] for music generation.

By contrasting metrics such as music quality, preference satisfaction, and descriptive correspondence in generated music, we believe it is desirable to effectively assess the music modeling capabilities, personalized music generation capabilities, and performance in balancing textual query and UMP of our proposed model.

Measurement of Personalized AMG.

- Music Quality: By assessing the musicality of generated music, we aim to evaluate the proposed music generation model's proficiency in music modeling and general music generation capabilities;
- Description Correspondence: Evaluating the alignment between generated music and input textual descriptions allows us to assess the proposed music model's understanding of textual queries and the effectiveness of their transformation into conditioning factors;
- Humanity: A primary concern with existing text-to-music generator models is their deficiency in accounting for the structural elements, repetitions, and motifs commonly considered by human composers during song composition [86]. Therefore, we evaluate the model's generated music in comparison to human compositions;
- Preference: Rating satisfaction with preferences enables the evaluation of the efficacy of UMP-aware AMG, while also assessing the model's ability to balance user preferences and textual queries in weighting constraints during music generation.

E. Participant Recruitment

We recruited the participants via email by advertising our research study to undergraduate and graduate students of National University of Singapore. In total, 50 volunteers registered for the study, 16 of them who claimed they were definite about their musical tastes were recruited for our study, and 15 completed the experiment end-to-end. The 1 participant who did not finish the study dropped off at the rating phase, which can be attributed to the significant user efforts required from listening to and rating the considerable amount of music files at this step. In summary, we collected valid responses from 15 participants, among whom 8 were female and 7 were male. Participants who completed the subjective experiment were rewarded 30 SGD (approximately 22.4 USD).

F. Experiment Procedures

To obtain a participant's U2A, U2B, and U2C, we create three graphs that connect a virtual user node with the participant's favorite music tags, songs, and subscribed users. We calculate U2D by averaging the first three UMPs and creates U_{rand} with random values. We obtain U1 using the baseline model $M1$ from the participant's preferred songs.

We evaluate the effectiveness of UMP models in music recommendation by predicting the likelihood of connections between each UMP and songs in our dataset. Participants listen to and rate the most and least recommended songs based on their music preferences (1-10 scale). U1 recommends songs based on cosine similarity between the UMP and song embeddings, as in the original work.

We evaluate the UMP-aware music generator and compare different combinations by generating songs using all combinations of six UMPs and three primers mentioned in subsection V-B. Participants are asked to rate the songs (1-10 scale).

We evaluate our text-to-music generation method by retrieving U2T from a graph that centers on a virtual node representing a random textual query built from a set of music tags. We predict the most and least recommended songs from U2T and U_{rand}. Participants will rate the songs based on their description correspondence to the textual query (1-10 scale).

We evaluate our UMP-aware text-to-music generation system by generating songs with 4 compared personalized AMG models, inputting the same user-specified text descriptions. Participants rate the songs based on music quality, description correspondence, humanity and preference satisfaction (1–10 scale).

During experiment, the UMP, primers and music generators where the songs come from are made invisible to participants. All songs are shuffled in order.

G. Results

UMP Capture and Incorporation: As shown in Table I, the results of music recommendation show that U1 and all U2s perform better than U_{rand}. This suggests that user information, music community relationships and listening histories can be used to effectively learn music preferences for recommendation and generation.

Compared to the baseline M_1, which relies solely on users' listening history, our proposed model improves recommendation and UMP-aware AMG by 31.3% and 63.5% respectively. The 31.3% improvement in recommendation is calculated as:

$$\left(\frac{R_{U2B}-R_{U1}}{R_{U1}}-\frac{NR_{U2B}-NR_{U1}}{NR_{U1}}\right)/2 \tag{11}$$

[4][Online]. Available: https://app.wavtool.com/

TABLE I
PREFERENCE RATINGS OF UMP MODELS' RECOMMENDATIONS AND PRIMER-UMP-GENERATED SONGS

UMP	Recommendation			Generation			
	R ↑	***NR*** ↓	**Incr.** ↑	P_1 ↑	P_2 ↑	P_3 ↑	**AVG** ↑
U2A	5.93 (1.25)	5.04 (2.90)	0.89 (2.44)	2.57 (0.62)	2.14 (0.30)	2.86 (1.68)	2.52 (1.04)
U2B	**7.20 (3.0)**	**4.52 (2.96)**	**2.68 (3.35)**	**3.86 (1.96)**	**4.00 (2.18)**	3.57 (1.77)	**3.81 (3.11)**
U2C	6.93 (2.80)	5.38 (1.74)	1.55 (2.81)	3.43 (1.50)	3.14 (1.44)	2.86 (1.37)	3.14 (1.50)
U2D	5.93 (1.23)	6.04 (0.42)	−0.11 (1.15)	2.57 (0.62)	3.43 (2.09)	**4.57 (3.63)**	3.52 (2.68)
U_{rand}	5.14 (/)	6.26 (/)	−1.12 (/)	2.29 (/)	2.00 (/)	2.00 (/)	2.09 (/)
U1	5.56 (0.67)	6.76 (1.04)	−1.2 (0.11)	2.14 (0.35)	2.14 (2.09)	2.71 (1.27)	2.33 (0.65)

The values within parentheses denote the t-test statistics based on the control group U_{rand} where n1=n2=15 thus the threshold is 2.13 at a confidence level of 0.05. (R: most recommended; NR: least recommended; Incr.: increment in ratings = R-NR; AVG: average.).
The colored values represent the best performance scores for each metric.

and the 63.5% improvement in UMP-aware AMG is calculated as $AVG_{U2B} - AVG_{U1}$. This demonstrates that listening history alone does not suffice to reflect a user's music preference. Also, M_1's average operation over all songs can result in an over-smooth modeling, which inaccurately describes personal preferences.

Among the proposed UMPs, U2B has the highest accuracy in capturing users' preferred and disliked song features by learning from selected songs. This can be attributed to to the extensive features and abundant song samples in our dataset. In contrast, U2D, which seeks to draw on the strengths of other UMP variants, performs poorly. This highlights the limitations of combining music preference embeddings from various sources, and suggests that future research should focus on a single modeling dimension to increase efficiency.

Moreover, U2B is also a superior approach for music generation, indicating that successful UMP modeling can guide personalized music generation effectively. However, the performance of various primer and UMP combinations varies, making it difficult to determine the impact of primers on personalized music generation. This can explain the less than ideal performance of the baseline due to its heavy dependency on seed songs, which is disadvantageous compared to neural network-based approaches.

T-tests were performed on subjective experimental outcomes for validation. As illustrated in Table I, the t-value for U2B group (proposed) versus U_{rand} group in the context of the "Recommendation Improvement" criterion stands at 3.35, markedly exceeding the critical t-value of 2.13 (where n1=n2=15) at a 0.05 confidence level. Moreover, the comparison of music generation ratings between these groups (Generation - AVG) yields a t-value of 3.11, likewise surpassing the threshold. In consequence, our subjective experimental results exhibit statistical significance in our subjective experimental results, affirming the effectiveness of our proposed model in capturing UMP and guiding AMG.

A collection of recommended and generated song demos of one participant can be found on the Soundcloud platform.

As stated in subsubsection V-A-2, we propose an offset likability factor to mitigate the impact of mainstream music preferences. If not addressed properly, popular songs can dominate music recommendations and limit the expression of individual music preferences, thus failing to provide personalized music recommendations and generation. In addition, we try a second method that manipulates song embeddings

TABLE II
PREFERENCE RATINGS OF UMP MODELS' RECOMMENDATIONS UNDER DIFFERENT OFFSETTING FACTOR c VALUES

c values	**0.0**	**0.1**	**1.0**	**5.0**	**10**	**20**
Preference Satisfaction	4.22	4.58	4.42	6.56	**7.03**	3.91

The colored values represent the best performance scores for each metric.

TABLE III
(A) SUBJECTIVE DESCRIPTION CORRESPONDENCE RATINGS OF AMG FROM QUERY-BASED UMP V.S. RANDOM UMP

	(a) Subjective Ratings		(b) Objective Scores		
Query	U2T	U_{rand}	(1) MusicCaps	(2) Last.fm	(3) U2T
Description Correspondence (DC)	**4.29 (2.89)**	2.43 (/)	0.85	1.03	**1.24**

The value within parentheses denotes the t-test statistics based on the control group U_{rand} where n1=n2=15 thus the threshold is 2.13 at a confidence level of 0.05. (b) objective relevance of human-written and query-based generated songs.
The colored values represent the best performance scores for each metric.

directly. Specifically, we obtain a weighted embedding e_{weighted} as $e_{\text{original}} - c * e_{\text{popular}}$, where e_{popular} represents the embedding of the most popular songs. As shown in Table II, a value of $c = 10$ minimizes the influence of popular songs, based on our experimentation with various values. It ensures that at least one top-K tag of the most recommended song corresponds to the user's selected tags/songs. We apply this method to $U2D$ and obtain its weighted version, $U2D'$, which achieved a better song recommendation score of $R = 7.03$ (+18.5% over R_{U2D}) among all tested c values. This shows that processing song embeddings directly is more effective in reducing the dominance of mainstream songs.

Text-to-Music Generation: As demonstrated in Table III-(a), our proposed system shows advantages in query-based music generation with a +76.5% improvement, calculated as

$$\frac{DC_{U2T} - DC_{U_{rand}}}{DC_{U_{rand}}} \tag{12}$$

The query-converted UMP embeddings are successfully applied to generate music that aligns well with the description. Similarly, we conducted a t-test, and its outcomes surpassing the threshold further substantiate the significance of this inference.

TABLE IV
SUBJECTIVE RATINGS OF SONGS GENERATED FROM COMPARED PERSONALIZED AMG MODELS

Music Generator	Music Quality ↑	Description Correspondence ↑	Humanity ↑	Preference ↑
BART	4.53	3.88	3.71	3.12
WavTool	5.06	3.18	4.23	3.65
Rule-based (U1)	**7.65**	2.29	**7.41**	2.53
Proposed (U2B+P_2)	6.35	**5.59**	6.12	**5.88**

The colored values represent the best performance scores for each metric.

Personalized AMG: As demonstrated in Table IV, our proposed model significantly outperforms comparative models in terms of description correspondence and preference satisfaction metrics, showing +44.1% and +61.1% improvement over the second best respectively. This highlights the efficacy of graph modeling in capturing both user musical preferences (UMP) and textual descriptions, striking a delicate balance between these two crucial aspects in music generation. Moreover, our model achieves commendable results in terms of music quality and Humanity metrics, indicating the superiority of smaller decoder-based models (i.e., Music Transformer) over larger general-purpose language models (e.g., BART, GPT-4) in symbolic music generation tasks. It is noteworthy that while integration of user textual descriptions into music generation remains unsuccessful, rule-based models exhibit the best performance in terms of music quality and humanity. This observation suggests that effective modeling and appropriate integration of rules within deep learning models could potentially enhance the performance of personalized music generation further. We leave this as an open problem for further investigation.

H. Meso Analysis

As part of our research in prompt engineering, we investigated the properties of textual queries that can guide a music generator to produce songs most aligned with their semantic meaning. Specifically, we categorized the textual queries provided by participants in terms of their length and relative average word frequency within the dataset. Subsequently, we compared the description correspondence ratings of songs generated for each category and visualized the data samples pairs of query length, rating and average word frequency, rating as scatter plots to analyze their distribution patterns.

As illustrated in Table V, for the proposed model, shorter queries composed of high-frequency words yield songs that best match the descriptions. This observation aligns with the results of the third-order fitting conducted on our dataset (as depicted by the green lines in Fig. 4 and Fig. 5). For instance, an example of a short and high-word-frequency query provided by a participant in the experiment is: "J-Pop in major scale. lively guitar riff in the foreground and a melodious string pad background.cheerful, happy, and soothing." This query is translated into three tags: [J-pop, guitar, beautiful]. Correspondingly, it achieves a high description correspondence score.

In contrast, such a feature is not reflected in other personalized AMG models. We attribute this phenomenon to the unique modeling approach adopted in graph-based models. The process of first converting text queries into tags for semantic modeling allows for a more precise capture of the core semantics of the query. However, the limited and predetermined number of tags in graph-based models makes them highly sensitive to query length and word frequency, presenting a trade-off.

TABLE V
OBJECTIVE TEXT-MUSIC RELEVANCE AND SUBJECTIVE RATING OF DESCRIPTION CORRESPONDENCE OF SONGS GENERATED FROM QUERIES WITH DIFFERENT LENGTHS/WORD FREQUENCY

Query Length	Short ≤20 words	Middle 20-40 words	Long ≥40 words
Objective Relevance	**2.26**	0.91	0.56
Subjective Rating	**7.33**	5.13	3.33
Average Word Frequency	Least 30% Frequent	Middle	Most 30% Frequent
Objective Relevance	0.67	0.85	**2.21**
Subjective Rating	5.20	5.60	**6.83**

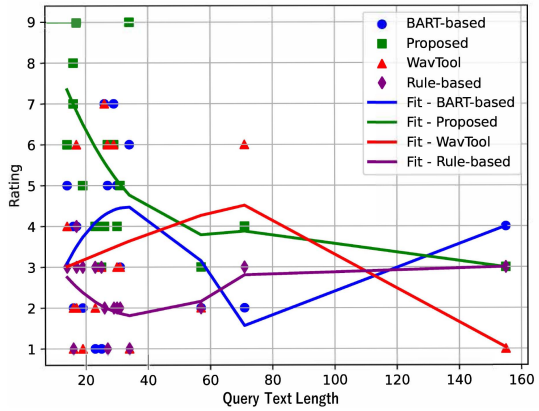

Fig. 4. Correlation between Descriptive Correspondence Ratings and Prompt Length in Compared Music Generation Models in Subjective Experiments, with Curves Representing Third-order Fit Results.

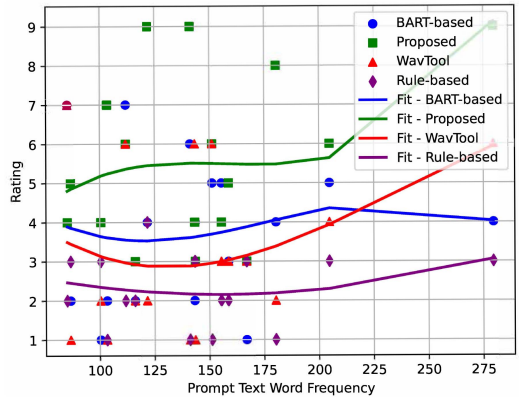

Fig. 5. Correlation between Descriptive Correspondence Ratings and Average Word Frequency of the Prompts in Compared Music Generation Models in Subjective Experiments, with Curves Representing Third-order Fit Results.

VI. Objective Experiment

Compared with UMP, evaluating the satisfaction of textual queries' embodiment in generated music is more quantifiable. We employ Mulan [50], a joint audio-text embedding, to measure the relevance between textual queries and the corresponding music. Mulan is trained on the MusicCaps dataset (following [11]), which consists of 5,521 audio files and their annotated captions. In our experiment, these captions are used as equivalents to our textual queries. We calculate and compare relevance scores for three music-query combinations: (1) {query, original music} from MusicCaps's test set, (2) {query, original music} from our collected dataset (as detailed in Section IV), and (3) {query, generated music} from our query-based music generation model. Our proposed AMG model has a 76.5% higher accuracy in matching queries, as shown in Table III-(b).

We speculate that our model converts queries into discrete tags, allowing the music generator to correlate the tags with the music strongly. Human annotations in setting (1) may differ in word selection, expression and features they concern, resulting in weaker correlation. Overall, our model can achieve human-level understanding of music descriptions and incorporate desired features into music generation.

Zooming into setting (3) {query, generated music} pairs mentioned above, the prompt engineering of text-to-symbolic music generation regarding query length and word frequency yields results similar to those observed in subjective experiments. We divide the pairs into smaller subsets based on query length and average word frequency of the query texts and re-evaluate the cross-modal relevance of these subsets. As shown in Table V, the higher cross-modal relevance scores suggest that shorter queries and high-frequency words have the highest correlation. This outcome reaffirms the efficacy of the proposed model in reducing the accessibility barriers for ordinary users who are unfamiliar with musical terms, by making it more effective and user-friendly thus allowing them to create music that suits their preferences and current needs.

VII. Discussion

Music generation is a complex task that requires a deep understanding of various musical concepts, such as rhythm, melody, and harmony. While general language models like ChatGPT have made significant progress in language tasks, it remains a challenge to apply them to highly specific tasks like music generation, which require the model to comprehend complex concepts from language inputs with few examples and to establish a correlation between generated samples and expected attributes.

To address this challenge, we propose a research direction that involves enabling large general language models to generate instructions that call on task-specific models. By doing so, we can seamlessly integrate language understanding and task execution, resulting in more accurate and personalized music generation. For example, instead of generating music by itself, ChatGPT can respond to a command like "music_generate -model huggingface/musicModelName -instruments [piano, guitar, vocal] -attributes genre: pop, tempo: 100, artist: Adele".

VIII. Conclusion

This study suggests a user music preference model that utilizes deep graph learning to capture fundamental music preferences from music communities in a multi-faceted and interpretable manner. It finds that textual queries can be accommodated into the graph to fine-tune the UMP, which further generalizes the model to accommodate multi-modal inputs. The experiment results demonstrate that the proposed UMP model is effective in guiding the generation of symbolic music from scratch that meets personal music preferences and matches textual queries. In summary, the study provides a solution to the increasing need for personalized music generation that promotes positive human activities.

Acknowledgment

The authors would like to thank anonymous reviewers for their valuable suggestions.

References

[1] D. Engh, "Why use music in english language learning? a survey of the literature," *English Lang. Teach.*, vol. 6, no. 2, pp. 113–127, 2013.
[2] D. Fisher, "Early language learning with and without music," *Reading Horizons: J. Lit. Lang. Arts*, vol. 42, no. 1, pp. 39–49, 2001.
[3] D. Murad, R. Wang, D. Turnbull, and Y. Wang, "SLIONS: A karaoke application to enhance foreign language learning," in *Proc. 26th ACM Int. Conf. Multimedia*, 2018, pp. 1679–1687.
[4] L. Nikol, G. Kuan, M. Ong, Y.-K. Chang, and P. C. Terry, "The heat is on: Effects of synchronous music on psychophysiological parameters and running performance in hot and humid conditions," *Front. Psychol.*, vol. 9, 2018, Art. no. 1114.
[5] A. L. Dvorak and E. Hernandez-Ruiz, "Comparison of music stimuli to support mindfulness meditation," *Psychol. Music*, vol. 49, no. 3, pp. 498–512, 2021.
[6] E. Kim et al., "Effects of exercise with music and physical contact on psychological states and interpersonal relationships," *Japanese J. Sport Psychol.*, vol. 41, no. 1, pp. 19–34, 2014.
[7] X. Ma, Y. Wang, and Y. Wang, "Content based user preference modeling in music generation," in *Proc. 30th ACM Int. Conf. Multimedia*, 2022, pp. 2473–2482.
[8] X. He, H. Zhang, M.-Y. Kan, and T.-S. Chua, "Fast matrix factorization for online recommendation with implicit feedback," in *Proc. 39th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2016, pp. 549–558.
[9] P. Dhariwal et al., "Jukebox: A generative model for music," 2020, *arXiv:2005.00341*.
[10] Mubert-Inc., "Mubert text-to-music," 2022. [Online]. Available: https://github.com/MubertAI/Mubert-Text-to-Music
[11] A. Agostinelli et al., "MusicLM: Generating music from text," 2023, *arXiv:2301.11325*.
[12] S. Wu and M. Sun, "Exploring the efficacy of pre-trained checkpoints in text-to-music generation task," 2022, *arXiv:2211.11216*.
[13] Y. Yu et al. ., "Lyrics-conditioned neural melody generation," in *Proc. MultiMedia Model.: 26th Int. Conf.*, 2020, pp. 709–714.
[14] C.-Z. A. Huang et al., "Music transformer," 2018, *arXiv:1809.04281*.
[15] F. Sanna Passino, L. Maystre, D. Moor, A. Anderson, and M. Lalmas, "Where to next? A dynamic model of user preferences," in *Proc. Web Conf.*, 2021, pp. 3210–3220.
[16] Y. Ma, L. Liao, and T.-S. Chua, "Automatic fashion knowledge extraction from social media," in *Proc. 27th ACM Int. Conf. Multimedia*, 2019, pp. 2223–2224.
[17] Y. Ma et al., "Knowledge enhanced neural fashion trend forecasting," in *Proc. 2020 Int. Conf. Multimedia Retrieval*, 2020, pp. 82–90.
[18] Y.-L. Chen, Y.-H. Yeh, and M.-R. Ma, "A movie recommendation method based on users' positive and negative profiles," *Inf. Process. Manage.*, vol. 58, no. 3, 2021, Art. no. 102531.
[19] A. A. Munaji and A. W. R. Emanuel, "Restaurant recommendation system based on user ratings with collaborative filtering," in *IOP Conf. Series: Mater. Sci. Eng.*, vol. 1077, no. 1, 2021, Art. no. 12026.

[20] C. Chen et al., "Personalized travel route recommendation algorithm based on improved genetic algorithm," *J. Intell. Fuzzy Syst.*, vol. 40, no. 3, pp. 4407–4423, 2021.
[21] S. Natarajan, S. Vairavasundaram, S. Natarajan, and A. H. Gandomi, "Resolving data sparsity and cold start problem in collaborative filtering recommender system using linked open data," *Expert Syst. with Appl.*, vol. 149, 2020, Art. no. 113248.
[22] Y. Hu, Y. Koren, and C. Volinsky, "Collaborative filtering for implicit feedback datasets," in *Proc. IEEE 8th Int. Conf. Data Mining*, 2008, pp. 263–272.
[23] V. Kant and K. K. Bharadwaj, "Enhancing recommendation quality of content-based filtering through collaborative predictions and fuzzy similarity measures," *Procedia Eng.*, vol. 38, pp. 939–944, 2012.
[24] S. Oramas, O. Nieto, M. Sordo, and X. Serra, "A deep multimodal approach for cold-start music recommendation," in *Proc. 2nd Workshop Deep Learn. Recommender Syst.*, 2017, pp. 32–37.
[25] A. Van den Oord, S. Dieleman, and B. Schrauwen, "Deep content-based music recommendation," in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, vol. 26, pp. 2643–2651.
[26] M. Jiang, Z. Yang, and C. Zhao, "What to play next? A RNN-based music recommendation system," in *Proc. IEEE 51st Asilomar Conf. Signals, Syst., Comput.*, 2017, pp. 356–358.
[27] X. Wang and Y. Wang, "Improving content-based and hybrid music recommendation using deep learning," in *Proc. 22nd ACM Int. Conf. Multimedia*, 2014, pp. 627–636.
[28] M. Gao et al., "Recommender systems based on generative adversarial networks: A problem-driven perspective," *Inf. Sci.*, vol. 546, pp. 1166–1185, 2021.
[29] E. Dervishaj and P. Cremonesi, "Gan-based matrix factorization for recommender systems," in *Proc. 37th ACM/SIGAPP Symp. Appl. Comput.*, 2022, pp. 1373–1381.
[30] A. B. Melchiorre and M. Schedl, "Personality correlates of music audio preferences for modelling music listeners," in *Proc. 28th ACM Conf. User Model. Adapt. Personalization*, 2020, pp. 313–317.
[31] D. J. Levitin, *This is Your Brain on Music: The Science of a Human Obsession*. NY, USA: Penguin, 2006.
[32] L. Zou et al., "Neural interactive collaborative filtering," in *Proc. 43rd Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2020, pp. 749–758.
[33] X. Zhao et al., "Recommendations with negative feedback via pairwise deep reinforcement learning," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2018, pp. 1040–1048.
[34] L.-C. Yang, S.-Y. Chou, and Y.-H. Yang, "MidiNet: A convolutional generative adversarial network for symbolic-domain music generation," *Int. Soc. Music Inf. Retrieval*, pp. 324–331, 2017.
[35] S. Tanberk and D. B. Tükel, "Style-specific Turkish pop music composition with CNN and LSTM network," in *Proc. IEEE 19th World Symp. Appl. Mach. Intell. Informat.*, 2021, pp. 000181–000185.
[36] Y. Huang, X. Huang, and Q. Cai, "Music generation based on convolution-LSTM," *Comput. Inf. Sci.*, vol. 11, no. 3, pp. 50–56, 2018.
[37] D. Eck and J. Schmidhuber, "A first look at music composition using LSTM recurrent neural networks," *Istituto Dalle Molle Di Studi Sull Intelligenza Artificiale*, vol. 103, p. 48, 2002.
[38] B. L. Sturm, J. F. Santos, O. Ben-Tal, and I. Korshunova, "Music transcription modelling and composition using deep learning," 2016, *arXiv:1604.08723*.
[39] D. Makris, M. Kaliakatsos-Papakostas, I. Karydis, and K. L. Kermanidis, "Combining LSTM and feed forward neural networks for conditional rhythm composition," in *Proc. Int. Conf. Eng. Appl. Neural Netw.*, 2017, pp. 570–582.
[40] H.-W. Dong, W.-Y. Hsiao, L.-C. Yang, and Y.-H. Yang, "Musegan: Multi-track sequential generative adversarial networks for symbolic music generation and accompaniment," in *Proc. AAAI Conf. Artif. Intell.*, 2018, vol. 32, no. 1.
[41] M. Akbari and J. Liang, "Semi-recurrent CNN-based VAE-GAN for sequential data generation," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2018, pp. 2321–2325.
[42] C.-Z. A. Huang et al., "An improved relative self-attention mechanism for transformer with application to music generation," 2018, *arXiv:1809.04281*.
[43] S. Di et al., "Video background music generation with controllable music transformer," in *Proc. 29th ACM Int. Conf. Multimedia*, 2021, pp. 2037–2045.
[44] Z. Dai et al., "Transformer-XL: Attentive language models beyond a fixed-length context," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics*, 2019, pp. 2978–2988.
[45] Z. Dai et al., "Transformer-xl: Language modeling with longer-term dependency," 2018.
[46] Y.-S. Huang and Y.-H. Yang, "Pop music transformer: Beat-based modeling and generation of expressive pop piano compositions," in *Proc. 28th ACM Int. Conf. Multimedia*, 2020, pp. 1180–1188.
[47] B. Yu et al., "Museformer: Transformer with fine-and coarse-grained attention for music generation," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 35, 2022, pp. 1376–1388.
[48] A. Muhamed et al., "Symbolic music generation with transformer-gans," in *Proc. AAAI Conf. Artif. Intell.*, 2021, vol. 35, no. 1, pp. 408–417.
[49] A. Radford et al., "Language models are unsupervised multitask learners," *OpenAI Blog*, vol. 1, no. 8, p. 9, 2019.
[50] Q. Huang et al., "MuLan: A joint embedding of music audio and natural language," 2022, *arXiv:2208.12415*.
[51] K. Zhao, S. Li, J. Cai, H. Wang, and J. Wang, "An emotional symbolic music generation system based on LSTM networks," in *Proc. IEEE 3rd Inf. Technol., Networking, Electron. Automat. Control Conf.*, 2019, pp. 2039–2043.
[52] H. H. Mao, T. Shin, and G. Cottrell, "DeepJ: Style-specific music generation," in *Proc. IEEE 12th Int. Conf. Semantic Comput.*, 2018, pp. 377–382.
[53] W. Su, Y. Fang, Z. Li, and X. Steven, "APE-GAN: A novel active learning based music generation model with pre-embedding," in *Proc. IEEE 4th Int. Conf. Electron. Inf. Commun. Technol.*, 2021, pp. 123–128.
[54] R. Yang et al., "Deep music analogy via latent representation disentanglement," *Int. Soc. Music Inf. Retrieval*, pp. 596–603, 2019.
[55] L. N. Ferreira and J. Whitehead, "Learning to generate music with sentiment," *Int. Soc. Music Inf. Retrieval*, pp. 384–390, 2019.
[56] C. Bao and Q. Sun, "Generating music with emotions," *IEEE Trans. Multimedia*, vol. 25, pp. 3602–3614, 2023.
[57] S. Sulun, M. E. P. Davies, and P. Viana, "Symbolic music generation conditioned on continuous-valued emotions," *IEEE Access*, vol. 10, pp. 44617–44626, 2022.
[58] J. Grekow and T. Dimitrova-Grekow, "Monophonic music generation with a given emotion using conditional variational autoencoder," *IEEE Access*, vol. 9, pp. 129088–129101, 2021.
[59] X. Tan, M. Antony, and H. Kong, "Automated music generation for visual art through emotion," in *Proc. 11th Int. Conf. Comput. Creativity*, 2020, pp. 247–250.
[60] E. Choi et al., "YM2413-MDB: A multi-instrumental FM video game music dataset with emotion annotations," *Int. Soc. Music Inf. Retrieval*, pp. 100–108, 2022.
[61] L. Ferreira, L. Lelis, and J. Whitehead, "Computer-generated music for tabletop role-playing games," in *Proc. AAAI Conf. Artif. Intell. Interactive Digit. Entertainment*, 2020, pp. 59–65.
[62] D. Lobo, J. Dcruz, L. Fernandes, S. Deulkar, and P. Karunakaran, "Emotionally relevant background music generation for audiobooks," in *Proc. IEEE Int. Conf. Artif. Intell. Mach. Vis.*, 2021, pp. 1–6.
[63] W. Wang et al., "CPS: Full-song and style-conditioned music generation with linear transformer," in *Proc. IEEE Int. Conf. Multimedia Expo Workshops*, 2022, pp. 1–6.
[64] H. Lim, S. Rhyu, and K. Lee, "Chord generation from symbolic melody using BLSTM networks," *Int. Soc. Music Inf. Retrieval*, pp. 621–627, 2017.
[65] H. Zhu et al., "Xiaoice band: A melody and arrangement generation framework for pop music," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2018, pp. 2837–2846.
[66] H. Bao et al., "Neural melody composition from lyrics," in *Proc. Natural Lang. Process. Chin. Comput.: 8th CCF Int. Conf.*, 2019, pp. 499–511.
[67] C. Zhang et al., "ReLYMe: Improving lyric-to-melody generation by incorporating lyric-melody relationships," in *Proc. 30th ACM Int. Conf. Multimedia*, 2022, pp. 1047–1056.
[68] H.-P. Lee, J.-S. Fang, and W.-Y. Ma, "iComposer: An automatic songwriting system for chinese popular music," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics (Demonstrations)*, 2019, pp. 84–88.
[69] Y.-J. Shih, S.-L. Wu, F. Zalkow, M. Muller, and Y.-H. Yang, "Theme transformer: Symbolic music generation with theme-conditioned transformer," *IEEE Trans. Multimedia*, vol. 25, pp. 3495–3508, 2022.
[70] Y.-C. Yeh et al., "Automatic melody harmonization with triad chords: A comparative study," *J. New Music Res.*, vol. 50, no. 1, pp. 37–51, 2021.
[71] Y. Teng, A. Zhao, and C. Goudeseune, "Generating nontrivial melodies for music as a service," *Int. Soc. Music Inf. Retrieval*, pp. 657–663, 2017.
[72] T.-P. Chen et al., "Functional harmony recognition of symbolic music data with multi-task recurrent neural networks," in *Proc. Int. Soc. Music Inf. Retrieval*, 2018, pp. 90–97.

[73] W. Yang, P. Sun, Y. Zhang, and Y. Zhang, "CLSTMS: A combination of two LSTM models to generate chords accompaniment for symbolic melody," in *Proc. IEEE Int. Conf. High Perform. Big Data Intell. Syst.*, 2019, pp. 176–180.
[74] R. Yang, T. Chen, Y. Zhang, and G. Xia, "Inspecting and interacting with meaningful music representations using VAE," 2019, *arXiv:1904.08842*.
[75] C. Raffel, "Learning-based methods for comparing sequences, with applications to audio-to-midi alignment and matching, Ph.D. dissertation, Columbia University, New York, NY, USA, 2016.
[76] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. 2019 Conf. North Amer. Chapter Assoc. Comput. Linguistics: Hum. Lang. Technol.*, 2019, pp. 4171–4186.
[77] H.-T. Hung et al., "EMOPIA: A multi-modal pop piano dataset for emotion recognition and emotion-based music generation," *Int. Soc. Music Inf. Retrieval*, pp. 318–325, 2021.
[78] N. S. Keskar, B. McCann, L. R. Varshney, C. Xiong, and R. Socher, "CTRL: A conditional transformer language model for controllable generation," 2019, *arXiv:1909.05858*.
[79] Z. Wang, D. Wang, Y. Zhang, and G. Xia, "Learning interpretable representation for controllable polyphonic music generation," *Int. Soc. Music Inf. Retrieval*, pp. 662–669, 2020.
[80] R. Madhok, S. Goel, and S. Garg, "SentiMozart: Music generation based on emotions," in *Proc. Int. Conf. Agents Artif. Intell.*, 2018, pp. 501–506.
[81] K. Choi, C. Hawthorne, I. Simon, M. Dinculescu, and J. Engel, "Encoding musical style with transformer autoencoders," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 1899–1908.
[82] H. H. Tan and D. Herremans, "Music FaderNets: Controllable music generation based on high-level features via low-level feature modelling," *Int. Soc. Music Inf. Retrieval*, pp. 109–116, 2020.
[83] V. P. Dwivedi and X. Bresson, "A generalization of transformer networks to graphs," 2020, *arXiv:2012.09699*.
[84] I. Simon and S. Oore, "Performance RNN: Generating music with expressive timing and dynamics," *Magenta Blog*, 2017.
[85] W. Zhu, T. Wen, G. Song, X. Ma, and L. Wang, "Hierarchical transformer for scalable graph learning," in *Proc. 32th Int. Joint Conf. Artif. Intell.*, 2023, pp. 4702–4710.
[86] S. Dai, H. Yu, and R. B. Dannenberg, "What is missing in deep music generation? A study of repetition and structure in popular music," *Int. Soc. Music Inf. Retrieval*, pp. 659–666, 2022.

Xichu Ma received the bachelor's degree from Sichuan University, Chengdu, China, in 2017, and the Ph.D. degree from National University of Singapore (NUS), Singapore, in 2023. He is currently a Research Fellow with the NUS School of Computing, NUS. His focuses on generative AI for human health and potential, particularly in music and lyrics generation. His work includes generating personalized and clinically preferred music for Parkinson's disease patients to aid in gait rehabilitation and creating educational lyrics to help children learn new vocabulary.

Yuchen Wang received the bachelor's and master's degrees in computer science from the National University of Singapore (NUS), Singapore, in 2022 and 2023, respectively. She is currently working toward the Ph.D. degree with the School of Computer Science and Engineering, Nanyang Technological University, Sinagpore. Her research focuses on the applications of Artificial Intelligence in multimedia generation, particularly in the fields of education and healthcare.

Ye Wang (Member, IEEE) received the B.Sc. degree from the South China University of Technology, Guangzhou, China, in 1983, the M.Sc. degree from the Braunschweig University of Technology, Braunschweig, Germany, in 1993, and the Ph.D. degree from the Tampere University of Technology, Tampere, Finland, in 2002. He is currently an Associate Professor with the Computer Science Department, National University of Singapore (NUS), Singapore. He established and directed the sound and music computing Lab (https://smcnus.comp.nus.edu.sg). Before joining NUS, he was a member of the technical staff with Nokia Research Center, Tampere, Finland, for nine years. His research philosophy is that technology should be developed for good - such as expanding access, increasing affordability, and improving quality of healthcare and education. Guided by this philosophy, he explored a new programmatic research agenda, which became his signature research in the past decade: cognitive neuroscience-inspired Sound and Music Computing for Human Health and Potential (SMC4HHP), attempting to address two big questions. 1) How to enable users to discover their preferred music that satisfies clinical requirements for Rhythmic Auditory Stimulation (RAS) based gait rehabilitation and exercise via music search, recommendation and generation? 2) How to leverage on the relationship between speech and singing to build applications for speech intervention? To address the above questions, he led the development of MusicRx technologies to make RAS accessible and affordable, and of SLIONS for speech intervention for various populations.