IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 16 February 2023, accepted 10 March 2023, date of publication 15 March 2023, date of current version 22 March 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3257381

RESEARCH ARTICLE

Hyperbolic Music Transformer for Structured Music Generation

WENKAI HUANG[1], **(Member, IEEE), YUJIA YU**[1], **HAIZHOU XU**[1], **ZHIWEN SU**[1], **AND YU WU**[2]

[1]School of Mechanical and Electrical Engineering, Guangzhou University, Guangzhou 510006, China
[2]Laboratory Center, Guangzhou University, Guangzhou 510006, China

Corresponding author: Yu Wu (wuyu8320@gzhu.edu.cn)

This work was supported by the Guangzhou Youth Science and Technology Education Project under Grant KP2023243.

ABSTRACT In the field of music generation, generating structured music is a highly challenging research topic. Music generation methods are currently learned in Euclidean space and usually modeled as a time series without structural properties, but due to the limitations of the time series representation in Euclidean space, the hierarchical structure of music is difficult to learn, and the generated music is poorly structured. Therefore, based on hyperbolic theory, this paper proposes a Hyperbolic Music Transformer model, which considers the hierarchy in music and models the structured components of music in hyperbolic space. Meanwhile, in order for the network to have sufficient capacity to learn music data with hierarchical and power regular structure, a hyperbolic attention mechanism is proposed, which is an extension of the attention mechanism in hyperbolic space based on the definition of hyperboloid and Klein model. Subjective and objective experiments show that the model proposed in this paper is able to generate high-quality music with structure.

INDEX TERMS Structured music generation, hyperbolic theory, hyperbolic attention, hyperbolic music transformer.

I. INTRODUCTION

Music is a form of artistic expression in which sound is composed and woven in a temporal sequence. The difference between music and random sound sources is that music has a complex structural hierarchy, and the perception of the structure of music is the basis of the listener's experience and interpretation of music; thus, the hierarchy is a fundamental aspect of the structural perception. The hierarchical structure of music refers to the long-term dependence, self-similarity, and repetition of music on multiple time scales. Lerdahl and Jackendoff [1], guided by professional music theory and psychology, scientifically classified it into four hierarchical structures: grouping structure, metrical structure, time-span reduction, and prolongational reduction. The grouping structure is the division of music into different-sized units, such as sections composed of phrases and sections composed of motives. The metrical structure is embodied in the beat; for example, Figure 1 shows that music is divided into phrases that consist of bars, and the bars consist of beats. This is the explicit hierarchical relationship, but music contains other hidden hierarchical messages, such as self-similarity or repetition between beats, which are the embodiment of the hierarchical structure of music represented as a tree in Figure 1. Time-span reduction indicates that music can be reduced to the most stable structure or tonic in terms of the time span, and this extraction process can be represented by a tree structure.

The interdependence, complexity, and richness of the hierarchical structure of music are some of its most important attributes. Deutsch and Feroe [2] found that listeners prefer music with a hierarchical pattern, and Lerdahl [3] found that a hierarchy helps people make connections between disparate pieces, so a hierarchy is necessary for listeners to enjoy music. The hierarchy of a musical piece has a direct impact on its overall quality and listeners' perceptions and evaluations of

The associate editor coordinating the review of this manuscript and approving it for publication was Pasquale De Meo.

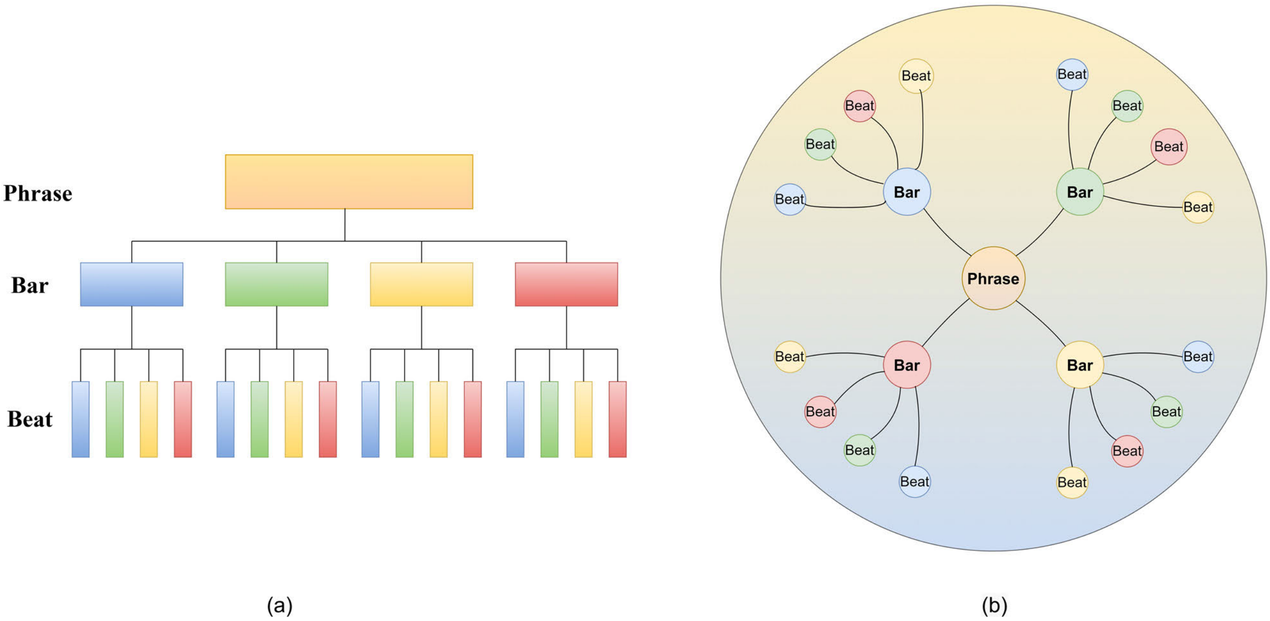


FIGURE 1. The rhythmic hierarchy of music can be represented as a tree, which is highly compatible with the negative curvature and exponential expansion of the hyperbolic space. (a) A schematic diagram of the tree structure: (b) A simple example of the hyperbolic space embedded in the tree structure. Those belonging to the lower level will be embedded in the edge of the circle, and those closer to the upper level will be embedded closer to the center of the circle.

the music; therefore, the ability to represent and understand the hierarchy of music is an important part of improving the quality of music generation.

Earlier, music generation was mostly based on music theory rules for simple pattern matching or statistical analysis-based methods, such as the Markov chain model for generating random notes with limited control, before modifying synthesized music. Steedman [4], on the other hand, generated musical sequences based on syntactic rule constraints. The first attempts to model music with neural networks also emerged with the development of technology [5], [6], [7]. In addition, with the development of deep learning, music generation models based on recurrent neural networks (RNNs) [8], [9] or convolutional neural networks (CNNs) [10] have also led to some progress in music generation. However, most models produce melodies with a dependence on the local temporal structure, spanning only a few bars and having a poor overall structure.

In recent years researchers have begun to focus on the importance of the musical hierarchy. Chu et al. [11] proposed a hierarchical recurrent neural network (HRNN), with a hierarchical and structural encoding of the composition added to the model as prior knowledge. The bottom layer generates melodies, while the higher layers generate drums and chords. MusicVAE [12] is a hierarchical variational autoencoder (VAE) model that focuses on learning long-term representations of music. Its latent vector is obtained by encoding the music sequence through an encoder and then decoding it in multiple steps, and such a decoder design allows MusicVAE to include certain structural information in the representation of music. Another tool, Music Transformer [13] established musical dependencies at the note level based on relative self-attention, expecting to learn the long-term structure of music. Meanwhile, MusicFrameworks [14] uses two transformer networks for a hierarchical musical structure representation and a multi-step generation process to generate full-length melodies guided by long-term repetitive structures, chords, melodic contours, and rhythmic constraints. Transformer-VAE [15], on the other hand, learns not only local and global features, but also to establish dependencies between vignettes, and it is capable of learning context-sensitive hierarchical representations. In addition, the Harmony-Aware Hierarchical Music Transformer (HAT) [16] proposes leveraging harmony-aware learning for structure-enhanced pop music generation. The model adaptively mines the structure of music and makes the musical tokens interact hierarchically to enhance the structure of multi-level musical elements. Further, an HRNN [17] consists of three long short-term memory (LSTM)-based sequence generators: a bar layer, beat layer, and note layer. The bar and beat layers are trained to generate bar and beat contours, which are used to represent the high-level temporal features of the melody. Meanwhile, the note layer is trained to generate melodies based on the bar and beat contour sequence outputs from the bar and beat layers. By learning at different time scales, the HRNN can grasp the general patterns of human melodies composed at different granularities and generate melodies with a more realistic long-term structure. Further, Pop Music Transformer [18] attempts to include a rhythmic structure in the input data so it can more easily learn the beat-bar-phrase hierarchy in the music. This data representation maintains the flexibility of local rhythmic variations and provides a way to control the rhythmic and harmonic structure of the music.

All these models improve the hierarchical structure and quality of the generated music to some extent. However, these models are chosen for use in Euclidean space, which ignores the limitations of modeling music with hierarchical properties in Euclidean space due to its narrow expressiveness and because it can only model data with polynomial growth. For music, the number of nodes grows exponentially with the number of layers, which will be severely distorted if it is embedded in a low-dimensional Euclidean space. Therefore, modeling music in Euclidean space is not a good way to learn the structure of music.

Recent studies have also shown that many types of complex data show highly non-Euclidean underlying anatomical structures [19], [20]. In such cases, Euclidean space does not provide the most robust or meaningful geometric representation. Work similar to [21], [22] has also shown that most data representations in machine learning applications lie on smooth manifolds, and because of this limitation in the ability of Euclidean spaces to model data with hierarchical structures well, many researchers have also turned their attention to combining machine learning with hyperbolic spaces in recent years [23], [24], [25] to find better representations of the hierarchical modeling of structural data.

Hyperbolic space follows a negative curvature, and it can be used for continuously learning hierarchical representations of text and graph data, and because hyperbolic space is smooth, it can be used for deep learning methods that depend on differentiability. Chen et al. [26] proposed a fully hyperbolic network based on the hyperbolic model, and it achieved better performance in natural language tasks. In addition, Gülçehre et al. [27] brought attention to hyperbolic networks, having achieved better results than when using Euclidean space in machine translation tasks. Zhang et al. [28], [29], [30] embedded a natural language processing task into hyperbolic space to learn the tree-like hierarchical structure of language, and it achieved good results. Music has some similarities with human language: expressions in language are built layer by layer, and their internal structure can be represented by a tree, where sentences are composed of words and paragraphs are composed of sentences — music takes on a similar structure. Inspired by the extension of hyperbolic space embedding to natural language processing and the image domain by [28], [29], [30], [31], and [32], in this paper, we embed music attributes into hyperbolic space to replace traditional Euclidean space music embedding, and we construct transformer networks in hyperbolic space to learn the hierarchical structure of music. We represent the data in hyperbolic coordinates, which are self-informative [28], so such a representation includes an explicit and an implicit hierarchy and it performs self-attentive operations in hyperbolic space, expecting that the network will learn the dependencies of, for example, bar and beat or the implicit relationships between notes to optimize the structure of the music.

The main contributions of the current paper are as follows:

(1) A novel structured music embedding strategy is proposed to obtain positive music expression by adopting the hyperbolic theory to encode the music sequences into the hyperbolic space.

(2) A novel Hyperbolic Music Transformer model with attention weight computation and aggregation is proposed to capture the hierarchical dependencies and generate the structured music.

(3) The proposed Hyperbolic Music Transformer model has been successfully applied to several subjective and objective experiments designed in this study. Experimental results demonstrate the proposed model Hyperbolic Music Transformer can generate high-quality music with structure, which provides a novel and effective method for the field of structured music generation.

II. METHODOLOGY

A. HYPERBOLIC THEORY

Hyperbolic space constitutes a negative constant curvature; it is a smooth Riemannian manifold, and it has several characteristics that Euclidean space does not: It can express the hierarchical structure and can embed the tree structure with low distortion. Further, the capacity of hyperbolic space embedding within the same upper bound is greater than that of Euclidean space. Therefore, the hyperbolic space cannot be embedded in the Euclidean space with equal measure, researchers have established many equivalent methods of modeling hyperbolic space. Commonly used models include the Hyperboloid model, Poincaré model, and Klein model. In this paper, we mainly apply the Hyperboloid and Klein models, which are conformal, so they can be easily mapped to each other, and based on the conformal feature, we can perform the attention weight aggregation in hyperbolic space.

1) HYPERBOLOID MODEL

The Hyperboloid model is commonly used for modeling n-dimensional hyperbolic spaces, and it is a manifold embedded in the n+1 dimensional Minkowski space. Figure 2 shows an example of the Hyperboloid model, where the n+1 dimensional Minkowski space is the real-valued space with an inner product of (1):

$$< u, v >_M = \sum_{i=1}^{n} u_i v_i - u_{n+1} v_{n+1} \tag{1}$$

The points in the n-dimensional Hyperboloid model can be represented by the points in the n+1-dimensional Minkowski space, given by:

$$\mathbb{H}^n = \left\{ x \in \mathbb{R}^{n+1} \mid < x, x >_M = -1, x_{n+1} > 0 \right\} \tag{2}$$

In the hyperboloid model, the distance between two points is given by

$$d_{\mathbb{H}}(u, v) = arccosh\left(-\langle u, v \rangle_M\right) \tag{3}$$

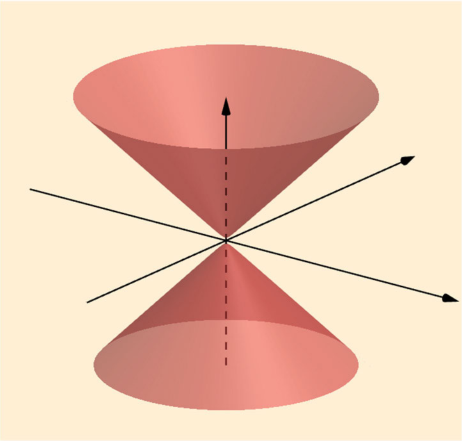


FIGURE 2. An example of a two-lobed hyperboloid.

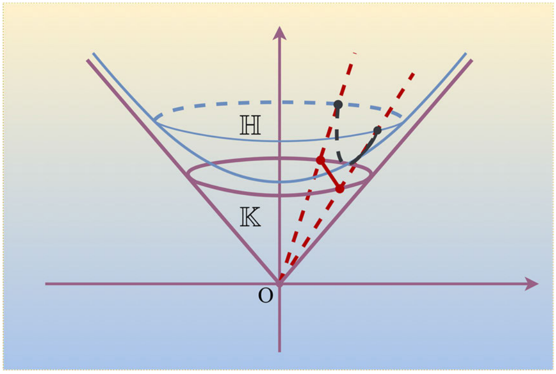


FIGURE 3. Example of the projection of the Hyperboloid model $\mathbb{H}^n$ with the Klein model $\mathbb{K}^n$.

2) KLEIN MODEL

The Klein model is obtained by mapping each point in $\mathbb{H}^n$ to the hyperplane $x_{n+1} = 1$ by the rays diverging from the origin. The Klein model is defined as:

$$\mathbb{K}^n = \{x \in \mathbb{R}^n | \, \|x\| < 1\} \tag{4}$$

where the plane past the origin intersects $\mathbb{H}^n$ to form the geodesic in the Hyperboloid model and the hyperplane with $x_{n+1} = 1$ to form the geodesic in the Klein model

Therefore, the Klein and Hyperboloid models can be easily mapped to each other, as shown in Figure 3, and the mutual mapping equation is given by:

$$\mathbb{H} \rightarrow \mathbb{K}\colon H2K\,(x)_i = \frac{x_i}{x_{n+1}} \tag{5}$$

$$\mathbb{K} \rightarrow \mathbb{H} : K2H\,(x) = \frac{1}{\sqrt{1 - \|x\|^2}}\,(x, 1) \tag{6}$$

where the distance in the Klein model can be calculated by converting it to the distance in the Hyperboloid model:

$$d_{\mathbb{K}}\,(u, v) = d_{\mathbb{H}}\,(K2H\,(u)\,, K2H\,(v)) \tag{7}$$

3) δ-HYPERBOLICITY

To verify the hypothesis of this study that the music dataset can be embedded in hyperbolic space, according to Khrulkov [32], we calculate whether the data are suitable for embedding in hyperbolic space by calculating the δ-hyperbolicity value. The calculated δ-hyperbolicity value after scaling is denoted as $\delta_H\,(X)$, $\delta_H\,(X) \in [0, 1]$, where the closer the value is to 0, the closer the dataset is to the hyperbolic space, and the calculation steps are as follows.

Let A be an arbitrary metric space with the distance function d, a, b, and $c \in A$, the Gromov product equations for a, b and c are given by:

$$(b, c)_a = \tfrac{1}{2}\,(d\,(a, b) + d\,(a.c) - d\,(b, c)) \tag{8}$$

For a set of points, the matrix G of pairwise Gromov products must be calculated. The δ value is then defined as the largest coefficient in the matrix $(G \otimes G) - G$, where $\otimes$ denotes the min–max matrix product, defined as:

$$A \otimes B = max_m \min \left\{A_{im}, B_{mj}\right\} \tag{9}$$

After obtaining the δ value and scaling it, we can calculate $\delta_H\,(X)$:

$$\delta_H\,(X) = \tfrac{2\delta(X)}{diam(X)} \tag{10}$$

where $diam(X)$ denotes the set diameter (maximal pairwise distance)

B. DATA REPRESENTATION

The representation used for the musical sequences of the input model in this study is revamped musical instrument digital interface (MIDI)-derived events (REMI) [18], which is a richer representation than MIDI-like. The REMI and MIDI-like representations have in common 128 Note-On events and Note Velocity. The difference is that, compared to MIDI-like, REMI uses Note Duration events instead of Note-Off events. REMI also adds Chord information to the musical representation, adding common chord roots and chord properties, so common chord events can be represented in combination. Other chord events can also be represented as needed. Further, REMI uses Position and Bar instead of Time-Shift, and Tempo events are added to each beat to indicate the beat-level velocity, as shown in Figure 4, which simply indicates the pentatonic notes with the corresponding REMI events. Such a beat-bar-phrase hierarchy is suitable for embedding into hyperbolic space learning, which is more appropriate for the needs of this study.

C. MODEL

1) HYPERBOLIC MUSIC TRANSFORMER

Transformer is a self-attentive sequence-based model that can capture long- and short-distance dependencies, and recently, Transformer has performed excellently in music generation tasks [13], [14], [18], so our team has considered the construction of the Hyperbolic Music Transformer model to

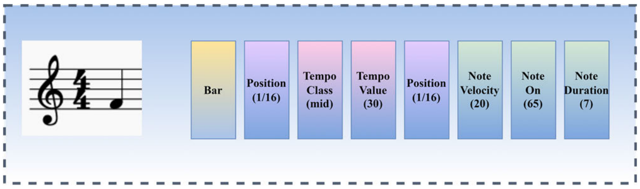


FIGURE 4. Example of a pentatonic note with a corresponding REMI event.

verify the hypothesis that music generation tasks are suitable for embedding hyperbolic spaces. Because the Hyperboloid model is the only unbounded model commonly used in hyperbolic space, it can easily embed data with arbitrary size parameters, and embedding is relatively stable, so we consider the Hyperboloid model to construct the Hyperbolic Music Transformer model. Figure 5 illustrates the main architecture of the Hyperbolic Music Transformer model, the core attention of which is presented in Figure 6.

Assuming the REMI word of the input model is $M \in \mathbb{R}^{\boldsymbol{n}}$, the feature matrix $X \in \mathbb{R}^{\boldsymbol{n}}$ is obtained by summing it with positional embedding after embedding, so the position coding formulas are as follows:

$$PE(pos, 2i) = sin\left(\frac{pos}{10000^{\frac{2i}{d_{model}}}}\right) \tag{11}$$

$$PE(pos, 2i+1) = cos\left(\frac{pos}{10000^{\frac{2i}{d_{model}}}}\right) \tag{12}$$

where d_{model} is the dimension of the word vector after word embedding.

The obtained feature matrix X is mapped to a matrix in the attention space by matrix multiplication:

$$\begin{aligned} Q &= XW_Q \\ K &= XW_K \\ V &= XW_V \end{aligned} \tag{13}$$

where W_Q, W_K, and W_V are the mapping matrices of the transformations of Q, K, and V, respectively, into the attention space.

To exploit the superiority of hyperbolic models in modeling hierarchical data, in this paper, we refer to the hyperbolic attention network of Gülçehre et al. [27] and consider constraining the latent variables on the hyperbolic model. The following operations are performed.

First, the computed attentions Q, K, and V are expressed in the polar coordinate form $((d, r) \in \mathbb{R}^{n+1}$:

$$\rho\left(Q\right) = \left(d_Q, r_Q\right) \tag{14}$$

where $\mathrm{r_Q} = \|\mathrm{Q}\|$ and $\mathrm{d_Q} = \frac{\mathrm{Q}}{\mathrm{r_Q}}$. Similarly, we can obtain $\rho\left(\mathrm{K}\right) = (\mathrm{d_K}, \mathrm{r_K})$ and $\rho(V) = (\mathrm{d_V}, \mathrm{r_V})$.

The mapping function θ is then used to constrain the polar coordinate form to a valid hyperbolic expression form, as shown in Figure 7:

$$\theta\left(\rho\left(Q\right)\right) = \left(sinh\left(r_Q\right) d_Q, cosh\left(r_Q\right)\right) \tag{15}$$

Similarly, the hyperbolic expressions $\theta(\rho(K))$, $\theta(\rho(V))$ of K, V can be obtained respectively.

The validity of the projection of this mapping function can be verified [28]:

(1) The hyperbolic parametrization of each point is equal to -1, so it can be verified by calculating $\langle Q_M, Q_M \rangle_M = -1$

(2) Calculate the distance between the point and the origin: if $d_{\mathbb{H}}\left(0, \left(d_Q, r_Q\right)\right) = r_Q$, then it can be verified that scaling is effective and the volume growth of the embedded space is exponentially related to the radius growth.

Then, the attention weights are calculated, the query vector q_i is taken from the query matrix Q, and the key vector k_j is taken from the key value matrix K. The transformer in the Euclidean space calculates the similarity by computing the dot product of the query vector q_i and the key vector k_j, the core of which is called scaled dot-product attention, given by:

$$Attention = softmax\left(\frac{QK^T}{\sqrt{s}}\right) V \tag{16}$$

Expressed as a vector expression:

$$attention_i = \sum_j \frac{\exp(a_{i,j})}{\sum_l exp(a_{i,l})} v_j \tag{17}$$

where $a_{i,j} = \langle q_i, k_j \rangle$.

Because hyperbolic space cannot measure the similarity directly using the dot product [19], this study refers to Gülçehre to measure the similarity of q_i and k_j in hyperbolic space using hyperbolic distance:

$$w_{qi,kj} = \exp\left(-\beta d_{\mathbb{H}}\left(\theta\left(\rho\left(q_i\right)\right), \theta\left(\rho\left(k_j\right)\right)\right) - bias\right) \tag{18}$$

where β and bias are parameters that can be set or learned by the network itself, and bias can be used to limit the non-negativity of the distance.

Because there is no natural definition of the mean on the manifold [19], the Einstein midpoint [33] is used instead of the weighted midpoint of the Euclidean space to perform the operation of aggregating the attention weights:

$$hypattention_i = \sum_j [\frac{w_{qi,kj}\gamma(v_{j\Bbbk})}{\sum_l w_{qi,kl}\gamma(v_{l\Bbbk})}] v_{j\Bbbk} \tag{19}$$

where $v_{j\Bbbk}$ is called the Lorentz factor, and it is given by:

$$\gamma\left(v_{j\Bbbk}\right) = \frac{1}{\sqrt{1 - \left\|v_{j\Bbbk}\right\|^2}} \tag{20}$$

where $v_{j\Bbbk}$ is a vector in the Klein space, and (5) can be used to map v_j on the Hyperboloid model to $v_{j\Bbbk}$ in the Klein space.

Substituting the Q, K, and V matrices into the computational attention, the original single-headed self-attention can be extended to a multi-headed parallel operation so the model can achieve better generalization, as multi-headed attention

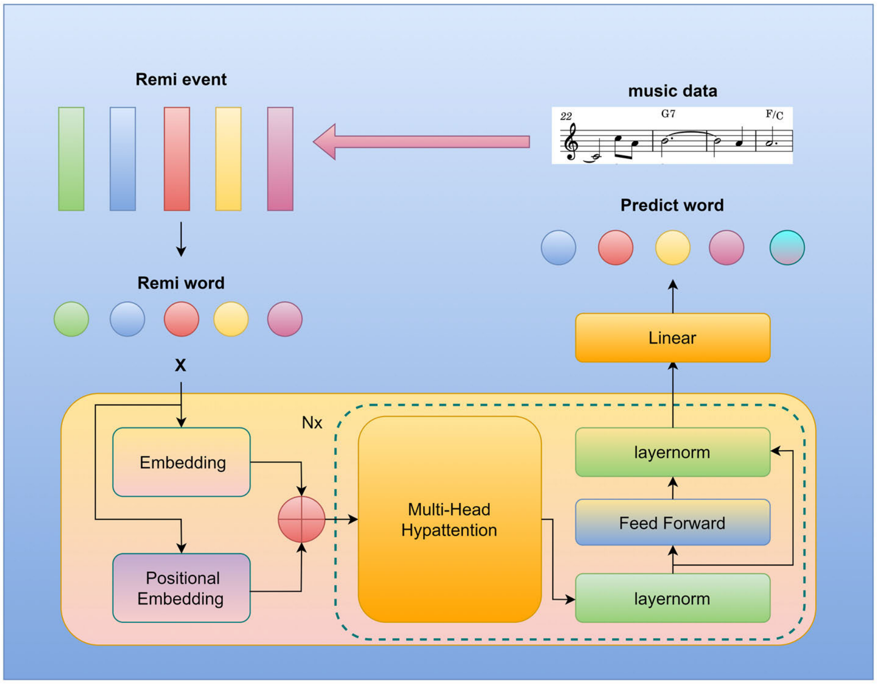


FIGURE 5. The architecture of the Hyperbolic Music Transformer model is used in this paper.

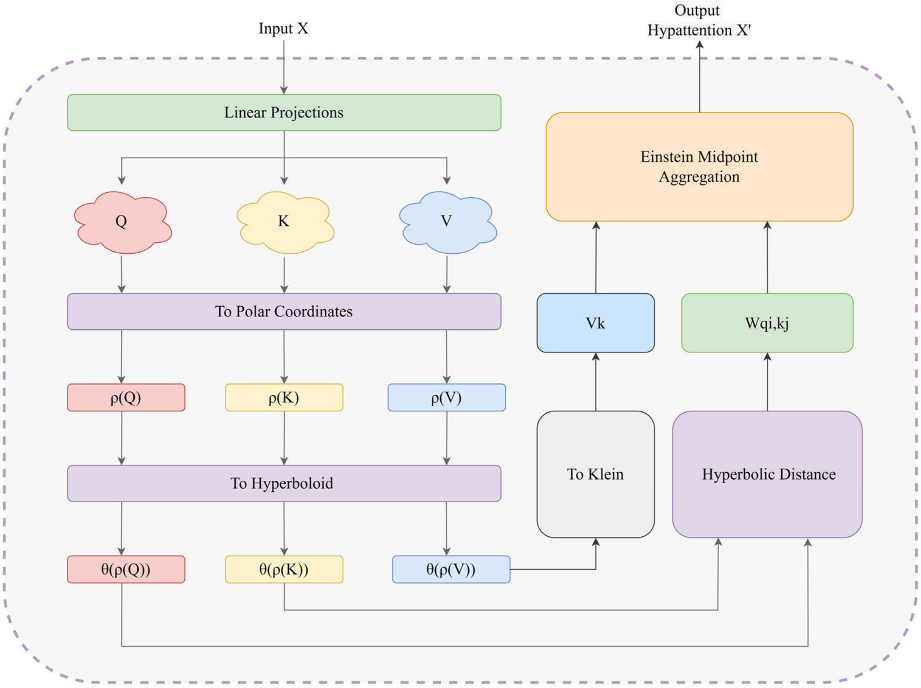


FIGURE 6. Details of the process of calculating attention in hyperbolic music transformer.

allows the model to attend to information jointly from different representation subspaces at different locations:

$$head_n = hypattention\,(Q, K, V) \tag{21}$$

$$Multi_{head} = concat\,[head, \ldots, head_h]\, W_O \tag{22}$$

where W_O is the output projection matrix and h is the number of heads.

2) HYPERBOLIC OPTIMIZATION

The commonly used optimizers for Euclidean space are designed to adjust parameters in Euclidean space that are

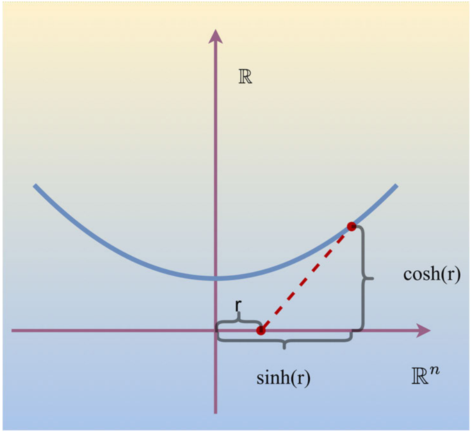


FIGURE 7. An example of the mapping function θ constraining the polar coordinate form to the hyperbolic expression form.

unsuitable for adjusting in non-Euclidean geometry, so the traditional optimizers are not adapted to the hyperbolic space models. Thanks to Riemannian stochastic gradient descent (RSGD) [34], the optimization problem of hyperbolic models is well solved. The optimizer uses an exponential map (Exp) to assign a Riemannian metric to the parameter space and performs a gradient update in the manifold to replace the regular update in the stochastic gradient descent (SGD) to optimize the hyperbolic space.

III. GENERATION

The trained model outputs probability vectors based on the input sequences, and the tokens are sampled to obtain the determined tokens; to make the generated samples more diverse, there are many sampling methods for music generation, not only the most intuitive greedy sampling (maximum probability sampling), but also common sampling methods, including temperature sampling, Top-k sampling, and Top-p sampling (Nucleus sampling).

A. TEMPERATURE SAMPLING

Temperature sampling is done by adding a hyperparameter t to the softmax function behind the output layer to normalize the output probability, and by adjusting t, the shape of the probability can be controlled to change, so the distribution of the lth token predicted is rescaled to:

$$p\left(x = u_l \mid x_{1:i-1}\right) = \frac{\exp(\frac{u_l}{t})}{\sum_{i \in V_l} \exp(\frac{u_i}{t})} \tag{23}$$

When t tends to 0, the distribution becomes sharp around the origin, which is similar to greedy sampling, and when t is larger, the sampled values are more random and interesting.

B. TOP-K SAMPLING

In Top-k sampling, the tokens are sampled from the Top-k tokens with the highest probability at each time step of decoding, and then the probabilities of these tokens are recalculated according to softmax; then, the next token is sampled according to the probability distribution.

C. TEMPERATURE-CONTROLLED STOCHASTIC SAMPLING METHOD WITH TOP-K

Due to the large randomness of the temperature sampling method, sometimes there may be an exceptionally low probability that a token is selected, and Top-k sampling can solve this situation by truncating the distribution of tokens in a more limited range, so the researcher can freely adjust the diversity by adjusting t and select the next token in the Top-k-truncated interval. The Top-k temperature sampling method is a trade-off between consistency and diversity.

In this paper, we use an improved temperature-controlled stochastic sampling method with Top-k [35] to generate music. This sampling method trusts the model distribution by approximating greedy sampling while avoiding repetition by penalizing. For a given list of generated tokens G:

$$p_{(x=u_l \mid x_{1:i-1})} = \frac{\exp(\frac{u_l}{t \cdot I})}{\sum_i \exp(\frac{u_i}{t \cdot I})} \tag{24}$$

where $l \in G, i \in G$, and $I(c) = \theta$ are customizable and equivalent to the above (24) when $\theta = 1$.

IV. EXPERIMENTS AND RESULT

In this section, experimental details and results will be presented. First, section A will introduce the dataset, section B will present the details of the word list obtained from the experiments, and then section C will present the results of the evaluation of data hyperbolicity. Then, section D will present the baseline model and model setup. Finally, the generated music will be evaluated. Our team performed objective and subjective evaluations, where section E will show the objective evaluation results using two methods: information rate and music generation evaluation (MGEval), and section F will show the subjective evaluation results, as well as arrange two sets of experiments with the specific experimental settings, and results will be listed in the corresponding subsections.

A. DATASET

This paper uses the POP909 dataset [36], which contains multiple versions of 909 piano arrangements of popular songs composed by professional musicians. The dataset contains vocal melodies, lead instrument melodies, and piano accompaniment for each song in MIDI format, as well as tempo, beat, key, and chord annotations.

B. DATA REPRESENTATION DETAIL

The experiment uses the REMI representation introduced in II. METHODOLOGY. In the POP909 dataset, there are 909 MIDI songs in total. First, we obtain the quantifiable music attributes needed for the REMI representation, and then we construct the vocabulary. The event tokens used to represent music in POP909 are shown in Table 1, where the bar is the symbol for the start of a bar, and for pitch, tempo,

TABLE 1. Number of events corresponding to the POP909 dataset after REMI encoding.

Event Type	Tokens
Bar	1
Position	16
Pitch (42–98)	55
Velocity (40–126)	44
Tempo (35–197)	47
Duration	17
Chord	121
EOS	1
All events	**302**

and velocity, some of these values do not have corresponding events. End of sequence (EOS) was also added to represent the end token, and its average token number was 1,658.

C. δ-HYPERBOLICITY RESULT

The hyperbolicity of the data was evaluated using the transformer model (introduced in section II-B). The calculated hyperbolicity measure is 0.26, which is close to 0 in the case of the POP909 dataset, indicating that the hypothesis of this study that the music data are suitable for embedding in hyperbolic space is supported.

D. BASELINES AND MODEL SETTINGS

To investigate the validity of extending the music generation task to hyperbolic space, this study aims to compare the music generated in different Euclidean and hyperbolic spaces and to use the REMI representation. Therefore, our team considered constructing the Euclidean Transformer using the REMI representation as a baseline model to compare with the Hyperbolic Music Transformer. For the Hyperbolic Music Transformer model, the data are mapped and activated into hyperbolic space, and the dependencies are captured in hyperbolic space using the attention mechanism (using Einstein midpoint aggregation), where the attention part of both the Euclidean Transformer and the Hyperbolic Music Transformer is set to 12 self-attention layers and eight attention heads. Each head is an independent unit of Q, K, and V. The d_{model} is set to 512, and the inner layer of the feed-forward part is set to 2,048. In addition, the learning rate is set to 1×10^{-4}, and the batch size is 10. Transformer in Euclidean space is optimized using Adam optimization, and the hyperbolic space transformer is optimized using RSGD. Both models were trained in NVIDIA RTX 3090 with 24-GB GPU, until the loss converged. After training, the initial sequence was randomly generated, and the model that was trained to convergence was selected to obtain the prediction sequence using the sampling method shown in section III. The sampling parameter t is set to 1.2, which was obtained after several experiments [35], and this setting allows a good balance between realistic and non-repetitive sampling. Then, the two models were set to generate sequences of equal length, and finally, they were converted to MIDI format.

E. OBJECTIVE EVALUATION

1) INFORMATION RATE

Music relies heavily on repetition to build structure, but the hierarchical structure of music is difficult to quantify. Objective experiments use information rate (IR) based on Variable Markov Oracle (VMO) [37] for comparison, as the IR itself can reflect the self-similarity structure in sequences [38], and IR values are larger when repetition and variation are balanced and smaller when sequences are random or highly repetitive. Therefore, a higher IR means there is a greater self-similarity structure in the sequence, which means more structured music; thus, the coherence and consistency of the music are higher.

Twenty samples were randomly selected from the music generated by POP909, the Euclidean Transformer, and the Hyperbolic Music Transformer, each with 512 tokens, and each MIDI sample generated was converted to WAV to calculate the IR values. Table 2 shows the average IRs of different samples, where the higher the IR, the more obvious the self-similarity structure. The IR of the original dataset is higher than the IR of the generated music, and the IR of the generated music in hyperbolic space is higher than the IR of the music generated in Euclidean space, indicating that the music generated in hyperbolic space is more structured. This shows that the Hyperbolic Music Transformer model proposed by this study can produce a higher level of consistency and a higher quality of musical structure.

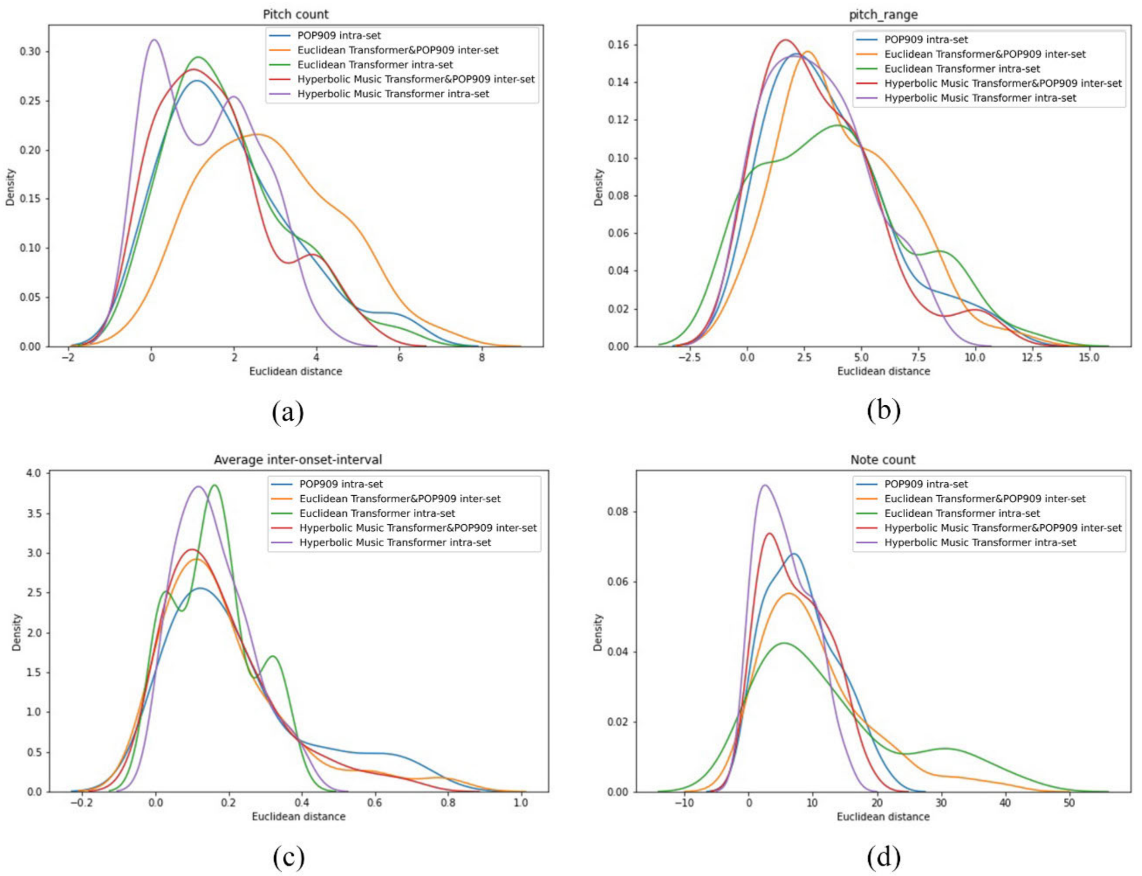
TABLE 2. The average IRs of different samples, where the high or low IRs can reflect the strength of the self-similarity structure.

Dataset	Total IR (Averaged scores)
POP909	13,352.85
Euclidean Transformer	11,683.18
Hyperbolic Music Transformer	12,417.58[1]

2) MGEVAL

MGEval is a toolbox designed by Yang [39] for the objective evaluation of music generation. It can extract features based on pitch and duration to measure how well the generated music matches the training data statistics. The extracted features are modeled as probability distributions, and the performance of the generated models can be represented by absolute and relative metric assessments, allowing the output of the music generation models to be evaluated and compared in an objective and reproducible manner.

The specific calculation process is shown in Figure 8. First, the features shown in Table 3 are extracted from the dataset, and then the absolute and relative metrics are evaluated. Absolute metrics are measures of the attributes and characteristics of a set of data, and they can be used to compare differences between the attributes of the training data set and the characteristics of the generated data or between the



(a)

(b)

(c)

(d)

FIGURE 8. An example of PDF of intra-set distance for POP909, Euclidean Transformer, Hyperbolic Music Transformer inter-set distance for Euclidean Transformer &POP909, inter-set distance for Hyperbolic Music Transformer& POP909, where (a) is pitch count (PC), (b) is pitch range (PR), (c) is average inter-onset interval (IOI), and (d) is note count (NC).

data generated by two different models. The relative measure allows the comparison of two distributions in different dimensions, first by applying two-by-two exhaustive cross-validation to calculate the distance from each sample to the same dataset (within a dataset) or to another dataset (between datasets) to obtain a distance histogram for each feature. The histogram is then smoothed using kernel density estimation to estimate the Probability Distribution Function (PDF) for each feature histogram.

In the objective evaluation experiment, our team randomly selected 20 music tracks each from the POP909 training dataset, the Hyperbolic Music Transformer, and the Euclidean Transformer, and then features are extracted from them to calculate their absolute and relative metric values. The relative metrics are the intra-set distance PDF (target distribution) from the training dataset and the inter-set distance PDF between the training dataset and the generated dataset, which are used to calculate the objective evaluation of the generated model: overlap area (OA) and Kullback–Leibler divergence (KLD). These two metrics can be used to measure the similarity and whether the model learns the features of the training set. Therefore, our team calculated the OA and KLD between the Hyperbolic Music Transformer & POP909 inter-set distances and POP909 intra-set distances, as well as between the Euclidean Transformer & POP909 inter-set distances and POP909 intra-set distances for comparison. The smaller the KLD and the larger the OA, the more similar the distribution of the two data sets are.

TABLE 3. The types of features in the MGEval used in the objective experiment.

Features		Introduction
Pitch-based features	Pitch count (PC)	The number of different pitches within a sample
	Pitch range (PR)	The pitch range is calculated by subtracting the highest and lowest used pitches in semitones
Rhythm-based features	Average inter-onset-interval (IOI)	The inter-onset-interval in the symbolic music domain.
	Note count (NC)	The number of used notes

From Figure 9 and Table 4, for the pitch count (PC), inter-onset interval (IOI), and note count (NC) features, the

TABLE 4. OA and KLD values for the distribution of the PC, PR, IOI, and NC of the generated music and original dataset, which can be used to measure similarities.

		Euclidean Transformer and POP909	Hyperbolic Music Transformer and POP909
PC	KLD	0.079	**0.072**
	OA	0.700	**0.712**
PR	KLD	**0.041**	0.103
	OA	0.791	**0.851**
IOI	KLD	0.389	**0.116**
	OA	0.758	**0.818**
NC	KLD	0.071	**0.001**
	OA	0.765	**0.810**

TABLE 5. The average PC, PR, IOI, and NC of the POP909 values are calculated from objective measurements, and the last two rows show the differences between the Euclidean Transformer and Hyperbolic Music Transformer relative to the POP909 training set. Using the POP909 as a benchmark, the smaller the difference, the better it is to learn the style of the original dataset.

	POP909	Euclidean Transformer	Hyperbolic Music Transformer
PC	10.6	-1.9	**-0.8**
PR	15.8	+1.9	**-0.2**
IOI	0.15	+0.36	**+0.34**
NC	104.5	+6.1	**+0.5**

KLD between the Hyperbolic Music Transformer & POP909 inter-set distances and POP909 intra-set is smaller than that between the Euclidean Transformer & POP909 inter-set distances and POP909 intra-set. Meanwhile, the OA between the Hyperbolic Music Transformer & POP909 inter-set distances and POP909 intra-set is larger than that between the Euclidean Transformer & POP909 inter-set distances and POP909 intra-set. For the pitch range (PR) feature, although the KLD between the Hyperbolic Music Transformer & POP909 inter-set and POP909 intra-set is larger than that between the Euclidean Transformer & POP909 inter-set and POP909 intra-set, the OA between Hyperbolic Music Transformer & POP909 inter-set and POP909 intra-set is larger than that between the Euclidean Transformer & POP909 inter-set and POP909 intra-set, showing that the performance of the two cannot be compared. In summary, it shows that the music generated by the Hyperbolic Music Transformer is more similar to POP909, the better the Hyperbolic Music Transformer is at learning the features of the original dataset compared to the Euclidean Transformer [40].

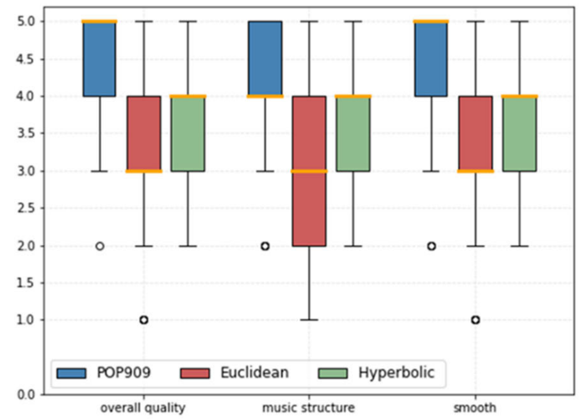


FIGURE 9. Visualization of the results of the experimenter's scoring of the overall music quality, structure, and fluency, where POP909 is blue, the Euclidean Transformer is red, and the Hyperbolic Music Transformer is green.

Following Yang [39], our team also conducted another comparison of an objective evaluation by first selecting 20 random MIDI tracks each from music generated by POP909, the Euclidean Transformer, and the Hyperbolic Music Transformer and evaluating them using the MGEval toolbox. Table 5 shows the measured average values of the PC, PR, IOI, and NC for the music generated by the POP909 dataset, Euclidean Transformer, and Hyperbolic Music Transformer, where the metrics of the music generated by the Hyperbolic Music Transformer are closer to those of the training set than those of the music generated by the Euclidean Transformer, indicating they are better able to learn the style and features of the original dataset.

F. SUBJECTIVE EVALUATION

Subjective evaluation experiments involved recruiting subjects to evaluate music from 18 tracks, including six tracks from among the model-generated tracks and six tracks from among the randomly selected POP909 dataset. The sample was randomly distributed before the experiment, and 15 participants in total were recruited, 11 male and four female. The number of participants with a musical background was five, representing one-third of the total.

In this experiment, participants were asked to rate each song from 1 to 5 (1 being the lowest and 5 being the highest score) on the following metrics:

(1) Overall quality of the music;
(2) Structure of the music;
(3) Smoothness of the music.

Figure 10 shows the experimental results. From the box plots, the median score of the POP909 dataset is five in terms of overall quality and fluency, four in terms of structure, and four in terms of the lower quartile, with the highest overall score. Second, the median scores of the music generated by the hyperbolic space model were four in overall quality, structure, and fluency, with the upper quartile being four and the lower quartile being three. The median scores of

TABLE 6. Statistical results of experiments for structural settings of music are shown as percentages.

	POP909	Euclidean Transformer	Hyperbolic Music Transformer
Did you feel the phrase?	96.6%	78.8%	83.3%
Smooth transitions between phrases?	85.5%	62.2%	68.8%
Does the pitch change harmoniously?	85.5%	65.5%	73.3%
Is the rhythm comfortable?	88.8%	58.8%	64.4%
Does music have a hierarchy?	90.0%	72.2%	83.3%
Music overall harmony?	90.0%	64.4%	72.2%
Are there abrupt notes?	18.8%	51.1%	36.3%

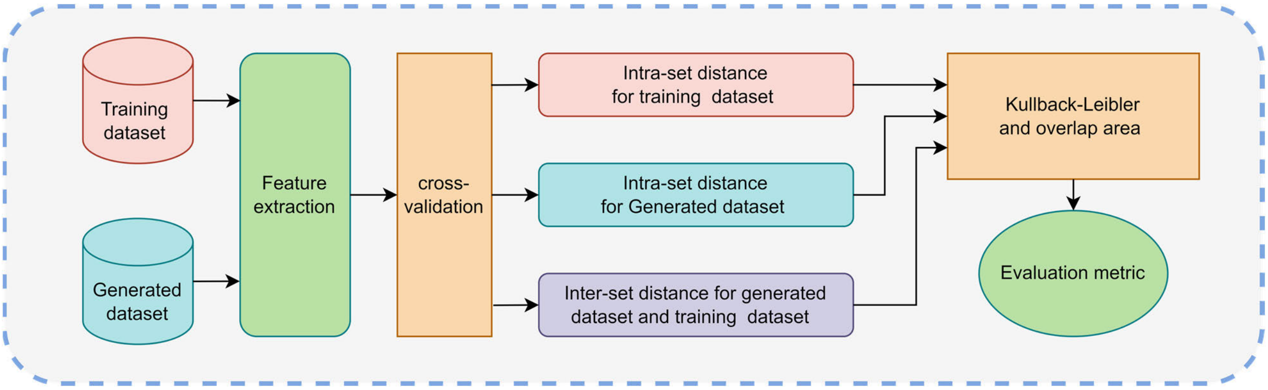

Training dataset
Generated dataset
Feature extraction
cross-validation
Intra-set distance for training dataset
Intra-set distance for Generated dataset
Inter-set distance for generated dataset and training dataset
Kullback-Leibler and overlap area
Evaluation metric

FIGURE 10. The process of calculating the intra-set distance, inter-set distance, KlD, and OA metrics in MGEval.

overall quality, structure, and fluency of the music generated by the Euclidean Transformer were all three, lesser than the median scores of the Hyperbolic Music Transformer, with the upper quartile being four. The lower quartile of overall quality and fluency is three, and the lower quartile of structure is two, which is smaller than that of the Hyperbolic Music Transformer model. Both the Hyperbolic Music Transformer and Euclidean Transformer models generated music with an upper edge of five and a lower edge closer to two, but the structural Hyperbolic Music Transformer model had a lower edge of two and the Euclidean Transformer model had a lower edge of one. In summary, the Hyperbolic Music Transformer model outperformed the Euclidean Transformer model in terms of the subjective ratings of the participants, especially in terms of structure.

Because the hierarchical structure of music is difficult to describe accurately, this study also set some related questions to evaluate the experiment specifically in terms of the structure of the music by referring to the literature [41], and the experimental question settings and evaluation results are shown in Table 6, with the same song settings as the previous experiment and the experimental options set to yes or no.

The results of the experiment are shown in Table 6, where the higher the proportion of yes choices for the first six questions, the better the musical hierarchy. From Table 6, we can see that the proportion of yes choices follows the pattern POP909> Hyperbolic Music Transformer > Euclidean Transformer, and the last question—"Are there abrupt notes?"— shows that the lower the proportion of yes choices, the smoother and more harmonious the music will be. From Table 6, the ratio of POP909 < Music Hyperbolic Transformer < Euclidean Transformer for the last problem selection can be obtained. In summary, it can be concluded that in terms of the music hierarchy, POP909> Hyperbolic Music Transformer > Euclidean Transformer. The Hyperbolic Music Transformer model in this paper has an advantage over the Euclidean Transformer model in terms of hierarchical structure.

V. CONCLUSION

In this paper, a novel structured music generation model, Hyperbolic music transformer, has been proposed. The purpose of this work is to generate music with hierarchy. Based on hyperbolic theory, our team encoded music in hyperbolic space to better represent the hierarchical structure of music. The hyperbolic attention proposed can capture hierarchical dependencies in the representation of music and these hierarchical dependencies can guide the generation of music with hierarchical structure. Extensive experiments demonstrated that the proposed model can generate high-quality music with structure and verified the feasibility of extending the music generation to hyperbolic space. In addition, although embedding music into hyperbolic space can be a significant way to learn the hierarchical information of music, it still deserves further research, especially for the pairwise relation between all bars in a melody. Our team will further investigate the hierarchical structure of music encoded in the form of directed graphs and embedded in hyperbolic space.

REFERENCES

[1] F. Lerdahl and R. Jackendoff, "An overview of hierarchical structure in music," *Music Perception, Interdiscipl. J.*, vol. 1, no. 2, pp. 229–252, Dec. 1983.

[2] D. Deutsch and J. Feroe, "The internal representation of pitch sequences in tonal music," *Psychol. Rev.*, vol. 88, no. 6, pp. 503–522, Nov. 1981.

[3] F. Lerdahl, "Tonal pitch space," *Music Percept.*, vol. 5, pp. 315–350, Jan. 2001.

[4] M. Steedman, "A generative grammar for jazz chord sequences," *Music Perception*, vol. 2, pp. 52–57, Oct. 1984.

[5] M. C. Mozer, "Neural network music composition by prediction: Exploring the benefits of psychoacoustic constraints and multi-scale processing," *Connection Sci.*, vol. 6, nos. 2–3, pp. 247–280, Jan. 1994.

[6] D. Eck, "A network of relaxation oscillators that finds downbeats in rhythms," in *Artificial Neural Networks—ICANN*. Berlin, Germany: Springer, 2001, pp. 1239–1247.

[7] D. Eck and J. Schmidhuber, "Learning the long-term structure of the blues," in *Proc. Int. Conf. Artif. Neural Netw.*, Berlin, Germany, Aug. 2002, pp. 284–289.

[8] J. Dai, S. Liang, W. Xue, C. Ni, and W. Liu, "Long short-term memory recurrent neural network based segment features for music genre classification," in *Proc. 10th Int. Symp. Chin. Spoken Lang. Process. (ISCSLP)*, Oct. 2016, pp. 17–20.

[9] H. Zhu, Q. Liu, N. J. Yuan, C. Qin, J. Li, K. Zhang, G. Zhou, F. Wei, Y. Xu, and E. Chen, "XiaoIce band: A melody and arrangement generation framework for pop music," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, London, U.K., Jul. 2018, pp. 19–23.

[10] L.-C. Yang, S.-Y. Chou, and Y.-H. Yang, "MidiNet: A convolutional generative adversarial network for symbolic-domain music generation," 2017, *arXiv:1703.10847*.

[11] H. Chu, R. Urtasun, and S. Fidler, "Song from PI: A musically plausible network for pop music generation," 2016, *arXiv:1611.03477*.

[12] A. Roberts, J. Engel, C. Raffel, C. Hawthorne, and D. Eck, "A hierarchical latent vector model for learning long-term structure in music," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 4364–4373.

[13] C. Z. A. Huang et al., "Music transformer," 2018, *arXiv:1809.04281*.

[14] S. Jin, Z. Gomes, and C. Dannenberg, "Controllable deep melody generation via hierarchical music structure representation," in *Proc. Int. Soc. Music Inf. Retr. Conf.*, 2021, pp. 1–8.

[15] J. Jiang, G. G. Xia, D. B. Carlton, C. N. Anderson, and R. H. Miyakawa, "Transformer VAE: A hierarchical model for structure-aware and interpretable music representation learning," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2020, pp. 516–520.

[16] X. Zhang, J. Zhang, Y. Qiu, L. Wang, and J. Zhou, "Structure-enhanced pop music generation via harmony-aware learning," 2021, *arXiv:2109.06441*.

[17] J. Wu, C. Hu, Y. Wang, X. Hu, and J. Zhu, "A hierarchical recurrent neural network for symbolic melody generation," *IEEE Trans. Cybern.*, vol. 50, no. 6, pp. 2749–2757, Jun. 2020.

[18] 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*, Oct. 2020, pp. 1180–1188.

[19] W. Peng, T. Varanka, A. Mostafa, H. Shi, and G. Zhao, "Hyperbolic deep neural networks: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 12, pp. 10023–10044, Dec. 2022.

[20] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst, "Geometric deep learning: Going beyond Euclidean data," *IEEE Signal Process. Mag.*, vol. 34, no. 4, pp. 18–42, Jul. 2017.

[21] J. M. Lee, *Introduction to Riemannian Manifolds*, 2nd ed. Berlin, Germany: Springer, 2019.

[22] J. M. Lee, *Introduction to Smooth Manifolds*, 2nd ed. New York, NY, USA: Springer, 2002.

[23] O.-E. Ganea, G. Bécigneul, and T. Hofmann, "Hyperbolic neural networks," 2018, *arXiv:1805.09112*.

[24] M. Nickel and D. Kiela, "Poincare embeddings for learning hierarchical representations," in *Proc. NIPS*, 2017, pp. 1–10.

[25] M. Nickel and D. Kiela, "Learning continuous hierarchies in the Lorentz model of hyperbolic geometry," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 3779–3788.

[26] W. Chen, X. Han, Y. Lin, H. Zhao, Z. Liu, P. Li, M. Sun, and J. Zhou, "Fully hyperbolic neural networks," in *Proc. 60th Annu. Meeting Assoc. Comput. Linguistics*, 2022, pp. 22–27.

[27] C. Gulcehre, M. Denil, M. Malinowski, A. Razavi, R. Pascanu, K. M. Hermann, P. Battaglia, V. Bapst, D. Raposo, A. Santoro, and N. de Freitas, "Hyperbolic attention networks," 2018, *arXiv:1805.09786*.

[28] C. Zhang and J. Gao, "Hype-HAN: Hyperbolic hierarchical attention network for semantic embedding," in *Proc. 29th Int. Joint Conf. Artif. Intell.*, Jul. 2020, pp. 3990–3996.

[29] M. V. Micic and H. Chu, "Hyperbolic deep learning for Chinese natural language understanding," 2018, *arXiv:1812.10408*.

[30] S. Dai, Z. Gan, Y. Cheng, C. Tao, L. Carin, and J. Liu, "APo-VAE: Text generation in hyperbolic space," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Human Lang. Technol.*, 2021, pp. 1–15.

[31] A. Ermolov, L. Mirvakhabova, V. Khrulkov, N. Sebe, and I. Oseledets, "Hyperbolic vision transformers: Combining improvements in metric learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 7409–7419.

[32] V. Khrulkov, L. Mirvakhabova, E. Ustinova, I. Oseledets, and V. Lempitsky, "Hyperbolic image embeddings," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 6418–6428.

[33] A. A. Ungar, *Analytic Hyperbolic Geometry: Mathematical Foundations and Applications*. Singapore: World Scientific, 2005.

[34] S. Bonnabel, "Stochastic gradient descent on Riemannian manifolds," *IEEE Trans. Autom. Control*, vol. 58, no. 9, pp. 2217–2229, Sep. 2013.

[35] 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*.

[36] Z. Wang, K. Chen, J. Jiang, Y. Zhang, M. Xu, S. Dai, X. Gu, and G. Xia, "POP909: A pop-song dataset for music arrangement generation," in *Proc. Int. Soc. Music Inf. Retr. Conf.*, 2020, pp. 1–8.

[37] C.-I. Wang and S. Dubnov, "Pattern discovery from audio recordings by variable Markov oracle: A music information dynamics approach," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2015, pp. 683–687.

[38] E. S. Koh, S. Dubnov, and D. Wright, "Rethinking recurrent latent variable model for music composition," in *Proc. IEEE 20th Int. Workshop Multimedia Signal Process. (MMSP)*, Aug. 2018, pp. 1–6.

[39] L. C. Yang and A. Lerch, "On the evaluation of generative models in music," *Neural Comput. Appl.*, vol. 32, no. 9, pp. 4773–4784, 2018.

[40] 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.

[41] R. Lang, S. Zhu, and D. Wang, "Pitch contours curve frequency domain fitting with vocabulary matching based music generation," *Multimedia Tools Appl.*, vol. 80, no. 18, pp. 28463–28486, 2021.

WENKAI HUANG (Member, IEEE) received the B.S. and M.S. degrees from Guangdong University of Technology, Guangzhou, China, in 2004 and 2007, respectively, and the Ph.D. degree from Guangzhou University, Guangzhou, in 2017. In 2007, he joined the School of Mechanical and Electrical Engineering, Guangzhou University, where he is currently an Associate Professor. His research interests include robot vision, medical image processing, and soft robotics.

YUJIA YU is currently pursuing the bachelor's degree in robotics engineering with Guangzhou University, Guangzhou, China. Her current research interests include deep learning (DL), music generation, electroencephalographic (EEG) signal analysis, and hyperbolic theory.

HAIZHOU XU is currently pursuing the bachelor's degree in robotics engineering with Guangzhou University, Guangzhou, China. His current research interests include deep learning (DL), electroencephalographic (EEG) signal analysis, and medical image processing.

ZHIWEN SU is currently pursuing the bachelor's degree in robotics engineering with Guangzhou University, Guangzhou, China. His current research interests include deep learning (DL), vision transformer (ViT), and hyperbolic networks.

YU WU received the B.S. degree from Gannan Normal University and the M.S. degree from Sun Yat-sen University. In 2006, she joined the Experimental Center, Guangzhou University, where she is currently an Experimenter. Her research interests include music generation, robot vision, and soft robotics.