International Journal on Semantic Web and Information Systems
Volume 20 • Issue 1

Enhancing Music Generation With a Semantic-Based Sequence-to-Music Transformer Framework

Yang Xu, Conservatory of Music, Shangqiu Normal University, China*

ABSTRACT

Music generation became a platform for creative expression, promoting artistic innovation, personalized experiences, and cultural integration, with implications for education and creative industry development. But generating music that resonates emotionally is a challenge. Therefore, we introduce a new framework called the Sequence-to-Music Transformer Framework for Music Generation. This framework employs a simple encoder-decoder Transformer to model music by transforming its fundamental notes into a sequence of discrete tokens. The model learns to generate this sequence token by token. The encoder extracts melodic features of the music, while the decoder uses these extracted features to generate the music sequence. Generation is performed in an auto-regressive manner, meaning the model generates tokens based on previously observed tokens. Music melodic features are integrated into the decoder through cross-attention layers, and the generation process concludes when "end" is generated. The experimental results achieve state-of-the-art performance on a wide range of datasets.

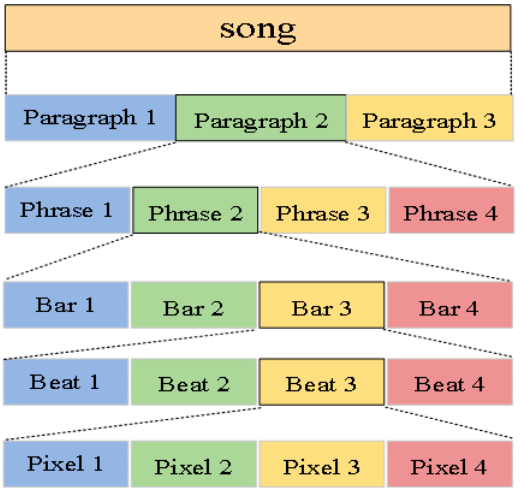
KEYWORDS

Deep Learning, Music Generation, Neural Networks, Transformers

Music, as an art form, involves the composition and arrangement of sounds over time, creating a tapestry of auditory experiences. Unlike random sources of sound, music distinguishes itself through its intricate and organized framework. This intricate framework forms a complex hierarchy, serving as the cornerstone for a listener's understanding and interpretation of music. In essence, this hierarchy becomes pivotal in how music's structure is perceived. At its core, the hierarchical structure of music encompasses several essential characteristics: its extended reliance on different time scales, the presence of self-resemblance, and the recurrence of patterns. Drawing from their expertise in music theory and psychology, scholars systematically classified this hierarchy into four distinct structures: grouping structure, metrical structure, time-span reduction, and prolongational reduction (Lerdahl & Jackendoff, 1983; Barbosa et al., 2022; Chen et al., 2022; Singh & Sachan, 2021). The grouping structure entails segmenting music into entities of various sizes, such as breaking down

DOI: 10.4018/IJSWIS.343491 *Corresponding Author

Figure 1. Hierarchical Arrangement Within a Musical Composition



compositions into phrases or further into motives. On the other hand, the metrical structure finds expression in the rhythmic beat. Figure 1 diagrams this structure, illustrating how music is divided into phrases composed of measures, and measures, in turn, are composed of beats. Although this explicit hierarchical arrangement is evident, music also conveys concealed hierarchical cues, such as instances of self-similarity or recurring patterns between beats. These subtleties form the embodiment of music's hierarchical structure, which is depicted as a tree in Figure 1. Time-span reduction unveils music's capacity to distill itself to its most stable and fundamental structure—the tonic—across different spans of time. This process of extraction lends itself well to representation through a tree structure, providing insight into how music evolves through its hierarchical layers.

The interconnected nature, intricate composition, and profound depth within the hierarchical arrangement of music constitute some of its paramount characteristics. Researchers (Ismail et al., 2022; Zhang et al., 2023; Li et al., 2019) discovered that audiences exhibit a preference for music imbued with a structured hierarchy. Moreover, Fred Lerdahl (Deutsch & Feroe, 1981) ascertained that this hierarchical framework facilitates the linkage of disparate components, thus underscoring its indispensable role in enhancing listeners' engagement with music. The hierarchical composition of a musical composition wields a direct influence over its holistic excellence and shapes listeners' perspectives and assessments of the musical piece. As a result, the capacity to effectively represent and comprehend the musical hierarchy assumes a pivotal role in advancing the quality of music generation.

The concept of algorithmic composition isn't novel. The origins of computational models for algorithmic composition can be traced back to as early as 1959 (Lerdahl, 2001), as corroborated by the research of Papadopoulos and Wiggins (1999) referred to by Zhang et al. (2021a). Using neural networks, even though of a shallower nature, for music generation dates back even further, to 1989. Yet, it wasn't until recent times, when deep neural networks showcased their prowess in comprehending extensive datasets, that music generation through neural networks gained significant traction. A considerable surge in the proposition of intricate deep neural network models for music generation has been witnessed over just the last couple of years.

The prevalent choice among current neural network architectures employed in music generation revolves around the recurrent neural networks (RNNs) and their derivatives. This inclination is driven by the innate nature of music generation, which fundamentally revolves around the creation of sequential patterns (Memos et al., 2018; Liu et al., April 2022; Nguyen et al., 2021; Yen et al., 2021; Papadopoulos & Wiggins, 1999). These models, although divergent in their underlying assumptions

and the methods they employ to represent and predict musical events, uniformly employ historical event data to influence the generation of the current event. Well-known instances of such models encompass the MelodyRNN variants, engineered for the generation of symbolic musical content (such as MIDIs), and the SampleRNN model (Mehri et al., 2016), tailored for generating audio-centric content (e.g., WAV files). Comparatively fewer endeavors have been directed toward harnessing the capabilities of deep convolution neural networks (CNNs) in the realm of music generation. However, a notable breakthrough arrived in the form of the WaveNet model (van den Oord et al., September 2016), a recent innovation tailored for audio-focused generation. This model takes a distinctive approach by generating individual audio samples sequentially. It achieves this through predictive distributions for each sample, conditioned on preceding samples via dilated causal convolutions. WaveNet's success highlights the potential for CNNs to engender authentic-sounding music. This development is quite promising, considering that CNNs are renowned for their faster training times and innate suitability for parallel processing in comparison to RNNs (Ilyas et al., 2022; van den Oord et al., June 2016).

Music Transformer (Ilyas et al., 2022; van den Oord et al., June 2016) serves as an additional tool that establishes interconnections within music, operating on the level of individual notes through relative self-attention. This approach seeks to imbue the model with an understanding of music's extended structural patterns. Conversely, MusicFrameworks (Huang, 2018) employs a tandem of transformer networks for delineating a hierarchical musical structure. This model's multistep generation process is engineered to produce comprehensive melodies, guided by overarching elements, such as long-standing repetitive patterns, harmonies, melodic shapes, and rhythmic constraints. In contrast, TransformerVAE (Dai et al., 2021) embraces a more holistic learning approach, encompassing both local and global attributes. It delves further into establishing correlations between distinct sections, fostering a contextually sensitive hierarchy of representations. Notably, the Harmony-Aware Hierarchical Music Transformer (HAHMT) (Jiang et al., 2020) introduces the concept of harmony-aware learning to augment pop music generation with enhanced structural attributes. This model adeptly mines the musical framework and orchestrates interactions between musical tokens across different levels, bolstering the multilevel structure of musical components. On a separate note, the HRNN model (Chopra et al., 2022) operates through three distinct sequence generators based on long short-term memory (LSTM): a bar layer, beat layer, and note layer. The bar and beat layers specialize in generating bar and beat contours, capturing the high-level temporal facets of melodies. Simultaneously, the note layer focuses on crafting melodies based on the bar and beat contour sequences derived from the preceding layers. This approach allows the HRNN to gain insights into the overarching patterns that shape human melodies at various scales, thereby generating melodies boasting a more authentic overarching structure.

Recent research has additionally highlighted that numerous intricate data forms exhibit intricate non-Euclidean foundational structures (Sarivougioukas & Vagelatos, 2022; Zhang et al., 2021b). This revelation underscores the inadequacy of Euclidean space in providing the most robust or meaningful geometric depiction in such scenarios. Comparable endeavors, such as those presented in Wu et al. (2020) and Peng et al. (2022), have established that the majority of data representations in machine learning contexts reside on smooth manifolds. This recognition of the limitations posed by Euclidean spaces in effectively capturing hierarchical structural data has led many researchers to pivot toward exploring the capabilities of Transformers (Bronstein et al., 2017; Lee, 2018). This shift is driven by the quest for more potent representations that align with the requirements of hierarchical modeling for structural data.

Regarding the above issues, we propose in this paper a new music generation framework based on the Transformer architecture, as shown in Figure 2. We convert the basic musical notes into a discrete token sequence and train the model to generate this sequence token by token. We employ a straightforward encoder-decoder Transformer architecture. The encoder extracts melodic features from the music, while the decoder uses these extracted features to generate the music sequence. Generation is performed auto-regressively; that is, the model generates tokens based on previously

observed tokens. At each step, the newly generated token value is fed back into the model to generate the next token value. We implement a causal mask on the self-attention module within the decoder to restrict tokens from attending to subsequent tokens. This causal masking mechanism ensures that the generation of a token at position i depends only on the preceding tokens at positions less than i. Melodic features of music are incorporated into the decoder via cross-attention layers. The generation process terminates when the "End" token is generated. The output sequence is directly used as the result. Experimental results demonstrate the effectiveness of our method's architecture, achieving state-of-the-art performance on multiple datasets. Note that all these existing methods heavily depend on meticulously crafted header networks and associated intricate loss functions (Lee, 2003; Zheng et al., 2022; Liu et al., May 2022; Lin et al., 2020). In contrast, our approach employs a simple encoder-decoder Transformer architecture with ordinary cross-entropy loss.

To summarize, our approach differs from previous approaches in terms of structural quality and structural space. For example, it is analyzed from the point of view of structural quality. First, we simplified the architecture by adopting a straightforward framework, enhancing the model's interpretability and training efficiency. Second, the model employs autoregressive generation, capturing temporal dependencies in music sequences and improving the coherence of the generated output. Third, the introduction of a causal masking mechanism helps maintain the sequence and logic in generation.

Analyzing from the perspective of structural space, we note that melodic features are integrated into the decoder through cross-attention layers, thereby enhancing the quality of the generated music. The mechanism to terminate the generation process upon generating an "End" token also prevents the generation of unnecessary or incomplete music segments. Finally, adopting a regular cross-entropy loss function simplifies the training process, improving the model's robustness.

In this paper we introduce a sequence-to-sequence learning approach for music generation, offering a fresh perspective by treating music generation as a generative task. We propose a pioneering structured music generation model, encoding music sequences into self-attention mechanisms by leveraging the encode and decode theory within the Transformer to create a meaningful musical representation. We also introduce a novel family of sequence-based music generation models that achieve a delicate equilibrium between speed and accuracy. Both objective and subjective experiments confirm the model's ability to produce high-quality structured music, providing an efficient method in the realm of music generation.

MATERIALS AND METHODS

Sequence Learning, Multitask Learning, Music Generation

In general, studies concerning music generation can be classified into three categories: sequence learning, multitask learning, and music generation.

Sequence Learning

Sequential data is prevalent in real-world datasets, such as speech, text, and stock predictions. In the last century, modeling problems related to sequences advanced substantially. Traditional methods such as Hidden Markov Models (Rezatofighi et al., 2019) have been widely used in fields such as text-to-speech conversion, language modeling, and protein sequences.

For example, Pierre Baldi and his colleagues used HMM to model proteins, adapting model parameters through algorithms that achieve smooth convergence. Simultaneously, Keiichi Tokuda employed an algorithm that generates speech parameters from HMM using unobservable vectors. However, traditional methods suffer from issues such as the need for manual feature design and extraction, leading to significant time and effort consumption. Therefore, deep learning has demonstrated outstanding performance in sequence modeling (Babalola et al., 2021; Liu et al., April

2023). It can model sequences in an end-to-end manner, avoiding the need for extensive manual features (Li et al., April 2022)—for instance, RNNs) and CNNs)

Although CNNs can address the problem to some extent, deep neural networks require more data to train a large number of parameters, and sometimes, there isn't even enough data for training. These issues have prompted consideration of alternative methods. For example, Ma et al. (2023) refers to Lample, who introduced an unsupervised machine translation method relying solely on monolingual corpora, and Liu, who employed SeqGAN to generate text using scarce, unmatched image-text data.

Multitask Learning

Multitask learning is frequently used to exploit shared features across interconnected tasks because features obtained from one task can be advantageous for others. Previous research has demonstrated the successful application of multitask learning across various domains of machine learning, spanning from natural language processing to computer vision (Liu et al., April 2023; Krauss, 2023). Zhang proposed enhancing generalization performance by using information from related tasks. Hashimoto established a hierarchical framework encompassing various natural language processing (NLP) tasks and formulated a basic regularization term to enhance performance across the board. Kendall adapted the relative weights for each task by formulating a multitask loss function aimed at maximizing Gaussian likelihood. There is still a substantial amount of ongoing work in the field of multitask learning (Zhou et al., 2023; Wu et al., 2023).

Music Generation

Over the past few decades, music generation has been a challenging task, and various approaches have been proposed (Pei et al., 2023; Shen, 2023). Typical data-driven statistical methods often employ Markov models. Additionally, other work has suggested similar ideas, such as using chords to select melodies. However, traditional methods require a significant amount of human effort and domain knowledge. Lately, deep neural networks have been employed for end-to-end music generation, effectively tackling the aforementioned challenges. Johnson, for instance, integrated a RNN with a non-recurrent neural network to depict the potential coexistence of multiple notes. (Moysis, February 2023) introduced an RNN-based generative model capable of generating four-part choral music using a Gibbs-like sampling process. In contrast to RNN-based models, Sabathe used VAE to learn the distribution of music pieces. Furthermore, Zhang employed a Transformer network (Moysis, July 2023) to generate music, using random noise as input to generate melodies from scratch.

Despite extensive research in music generation, none of the studies have fully considered the specificity of music, such as chords, rhythm, and instruments. For the generation of pop music, prior works did not take into account chord progressions and rhythmic patterns. Specifically, chord progressions typically guide the melody's progression, and rhythmic patterns determine whether a song is suitable for singing. Furthermore, pop music should also retain the characteristics of instruments. Finally, harmonies play a crucial role in multitrack music, but have not been well addressed in previous research. Moreover, music style is an essential feature of music. Recently, researchers have shown increasing interest in this area. An unsupervised music style transfer method has been proposed that does not require parallel data. This method is suitable for waveform and image data, but it cannot handle sequential data, such as Musical Instrument Digital Interface (MIDI) files. To address this issue, a variational autoencoder neural network model has been designed to achieve style transformation between classical and jazz music. Although this model can handle sequential data, it requires a significant amount of parallel music data for training. Therefore, the valuable question of how to leverage unparalleled music data to learn music styles remains.

Our sequence learning framework embodies a similar ethos to that of Pix2Seq (Zhang, 2023). Both methods view their domain tasks as sequence generation challenges and discretize the sequences' continuous values into integers. However, our approach diverges from Pix2Seq in three key aspects:

- Sequence structure: Pix2Seq sets up sequences using object coordinates and object categories, whereas our method uses basic musical notes.
- Architecture: Pix2Seq employs ResNet (Chen et al., 2021) as its backbone network, followed by an encoder-decoder transformer. Our approach is simpler and more direct, using a single encoder and decoder Transformer. It uses BERT (He et al., 2016) as the encoder to extract features and adopts causal transformer blocks as the decoder for sequence generation.
- Task: Pix2Seq is tailored for computer vision, whereas our approach is tailored for music generation.

Models

In this section we provide a detailed explanation of the proposed Sequence-to-Music Transformer Framework for Music Generation (Seq-Music) method. First, we provide a brief overview of our music composition framework. Next, we outline the transformation from musical notes to sequences. Finally, we describe the training and inference processes.

Overview

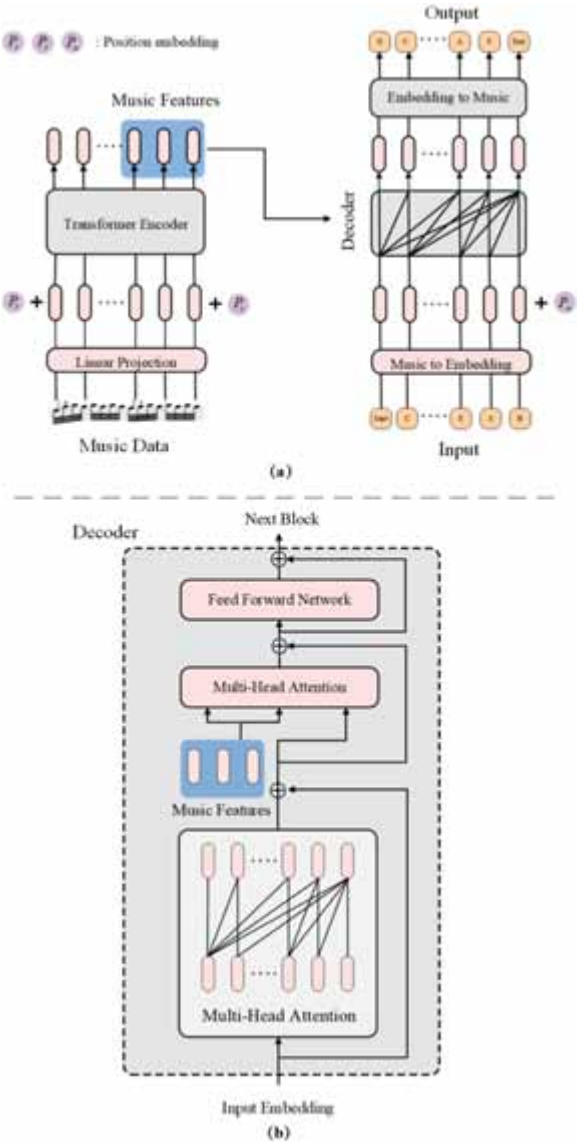
We present the architecture of the Seq-Music method as shown in Figure 2. It primarily consists of a straightforward encoder-decoder transformer structure. Musical notes are first transformed into a series of discrete tokens—for example, [C, D, E, F, G]. The encoder extracts features from the input music. The decoder, using the features extracted by the encoder, autoregressively generates the target music. To ensure tokens attend only to preceding tokens, we incorporate a causal attention mask into the self-attention module within the decoder. In addition to the basic musical note tokens, we introduce two special tokens (Start and End). These special tokens signify the start and end of music generation, respectively. During training, the input tokens for the decoder are [Start, C, D, E, ...], and the output sequence is [C, D, E, ..., End]. During inference, the decoder's input tokens begin with the start token. In each subsequent iteration, a fresh musical note token is generated and appended to the input tokens for the subsequent step to produce the next musical note token. The generation process concludes once the music composition is complete.

Note Segment Representation

The musical note template segment t represents segments with a better melody, and segment s indicates segments to be observed carefully in the subsequent music. In many existing methods, the segments of interest are often much smaller than the music segments being searched for later. We divide the music search segments and template segments into patches $s_p \in R^{N\times P^2\times 3}$ and $t_p \in R^{N\times P^2\times 3}$, where $N = \frac{HW}{P^2}$ represents the patch number. We then employ a linear projection to map the music segments to music embedding and add position embedding to the patch embedding to retain positional information. Subsequently, these fused embeddings are fed into the encoder.

We convert musical notes into a series of discrete tokens. Specifically, the generated music consists of basic notes [C, D, E, F, ...]. Notes can be composed in various ways, and we use the [C, D, E, F] composition because it aligns better with prior knowledge. Similar to our learning process, where we primarily learn notes starting from C, each note is uniformly discretized into $[1, n_{bins}]$. We use a shared vocabulary V for all notes because there are 88 keys on a piano. In our experiments, n_{bins} is set to 88. At the same time, to ensure that the output of each musical note element depends on the preceding musical note element, a mask is also added to the final input. Through an attention mask, the output embeddings at position i are constrained to focus exclusively on the input embeddings from positions before i, as shown in Figure 2(a). Then, feature extraction from multiple heads in the attention mechanism is integrated into word embeddings. This enables word embeddings to focus on

Figure 2. The Structure of the Proposed Seq-Music (a) and the Detailed Decoder Block (b)



the features extracted by the encoder. Finally, the next block's embeddings are generated through a feed forward network (FFN).

Training and Inference

Training. Our approach bears resemblance to language modeling (Mushtaq & Cabessa, 2023), where we maximize cross-entropy between preceding sub-tokens and the conditional log-likelihood given the input musical notes. The formula is shown in equation (1).

$$maximize \sum_{j=1}^{L} \log Q(\hat{z}_j \Big| s, t, \hat{z} < j) \tag{1}$$

In this equation, j represents the token's position, L is the length of the music generation, $Q(\cdot)$ represents softmax probability, and $Z < j$ is used to predict the preceding subsequence of the current token Z_j. The input sequence is a target sequence with musical notes (beginning and ending tokens omitted), as shown in Figure 2(b). Combining such tokens with causal masking helps maintain the autoregressive nature of the music generation model. Music generation can be seen as a description of musical notes, and the training objective is to generate subsequent music based on preceding musical notes using our approach.

Inference. During the inference process, the encoder perceives the sequence of musical notes that follow. The decoder's initial input is the start token, which serves as a special token to instruct our model to begin generating music. Subsequently, we generate musical note tokens one by one. Therefore, our model selects samples from the vocabulary V through maximum likelihood analysis, as determined by Equation (1). Additionally, to enhance the richness of music generation, extensions can be applied. During inference, prior knowledge can be leveraged to ultimately improve the model's completeness and richness.

RESULTS

In this section, the main focus is on the experiment details and results. First, we introduce the dataset and then we provide detailed information about the word list obtained from the experimental dataset. Next, we discuss our model setup and the baseline model. We then conduct both objective and subjective analyses of the generated music. In our experiments, we assess the results using two methodologies: information rate and music generation evaluation. Finally, we perform a sensitivity analysis.

Dataset

Our experimental datasets consist of the POP909 dataset (Paul et al., 2022), and the Lakh MIDI dataset (Wang et al., 2020). The POP909 dataset comprises 909 piano compositions created by professional musicians, including multiple versions. It also includes vocal melodies, piano accompaniments, primary instrumental melodies, and rhythmic components in MIDI format. We chose to perform our experiments using the extensively used Lakh MIDI dataset, which comprises music featuring multiple instruments in MIDI format. Following the processing steps, our ultimate dataset encompasses 29,940 songs, equivalent to 1,727 hours of music, with an average of 95 bars per song. The POP909 dataset is commonly used for research and experiments in the fields of music information retrieval, music generation, and machine learning.

The Lakh MIDI dataset is a massive dataset of MIDI files covering a wide range of musical styles and genres. The dataset contains a large number of MIDI files from the internet, totaling hundreds of thousands of songs. These MIDI files cover a wide range of music genres from classical to pop music. The Lakh MIDI dataset is commonly used for music generation, music analysis, computer musicology research, and music projects related to deep learning. Because of its diversity, researchers can use this dataset for a wide range of experiments and studies.

Construct Vocabulary

First, through our method on the POP909 dataset, we extract the required quantifiable music attributes to construct a vocabulary. Table 1 displays the event labels representing music in the POP909 datasets. In Table 1, "Start" denotes the starting symbol. Additionally, we have added "End" to represent the end token.

Table 1. Event Count Associated With POP909 Post REMI Encoding

Event type	Tokens
Start	1
Bar	1
Position	16
Tempo	47
Velocity	44
Pitch	55
Duration	17
Chord	121
End	1
All events	**303**

Baselines and Model Settings

Our method is implemented in PyTorch. During training, the batch size is set to four songs. In our experiments, we use the Adam optimizer with parameters β_1, β_2, and ε set to 0.9, 0.98, and 10^{-9}, respectively. The learning rate linearly ramps up to a peak of 10-4 and then decreases in inverse proportion to the square root of steps. The weight decay for regularization is set to 0.01. During the inference process, we use $top-k$ sampling with k set to 8. The generation continues until either the end token is generated, or the maximum length is achieved.

Objective Evaluation and Subjective Evaluation

Objective Evaluation

Information rate. In many pieces of music, repetitive structures are quite common, and this is even frequently observed in classical music. However, quantifying the hierarchical structure of music can be challenging. In objective experiments, a comparative analysis is conducted using information rate. This choice is rooted in the fact that information rate can reflect self-similarity in sequences (Choi et al., 2020). When a balance between repetition and variation exists, the information rate tends to be higher. Conversely, when there is high repetition in the sequence or the sequence appears random, the information rate tends to be lower. Therefore, a higher information rate indicates the presence of significant self-similarity structures in the sequence, indicating a level of structure and coherence. Hence, it can be inferred that the generated music exhibits higher consistency and coherence. Forty examples were randomly selected from the music generated by Music Transformer, Longformer, and Euclidean Transformer. Each example contains 1,024 tokens. Furthermore, each generated MIDI sample is converted into WAV format, and the information rate values are calculated, as shown in Table 2. Table 2 lists the average information rates for different samples. A higher value indicates more pronounced self-similarity structures, highlighting that our method generates more structured music. This information underscores our method's ability to generate music with clearer structural patterns and higher consistency.

As shown in Table 2, our method shows a clear advantage in overall information. Compared with other methods, our method has an average score of 15,201.81, which is much higher than other methods. Specifically, compared with Museformer, Euclidean Transformer, Hyperbolic Music Transformer, Music Transformer, Longformer, and Linear Transformer, our method performs better in

Table 2. The Mean Interference Ratios (IRs) Across Various Samples, With High or Low IR Values Serving as Indicators of the Self-Similarity Structure's Degree of Strength

Methods	Total IR (averaged scores)
Museformer (Wu et al., 2023)	14,251.12
Euclidean Transformer (Pei et al., 2023)	12,345.65
Hyperbolic Music Transformer (Shen, 2023)	13,895.73
Music Transformer (Moysis, February 2023)	13,981.42
Longformer (Moysis, July 2023)	14,654.38
Linear Transformer (Zhang, 2023)	14,166.54
Our	15,201.81

terms of overall performance. This result not only reflects the efficiency and reliability of our method but also shows that our method is highly stable. Therefore, these results demonstrate the significant advantages of our method and provide better support for our research.

MGEVAL. A music generation tool for objective evaluation called MGEVAL, designed by Yang (Huang et al., 2023), has been used. This tool can extract features from both the pitch and rhythm of the music, evaluating the similarity between the training data and the generated music. It then models the extracted features using probability distributions and evaluates the model's efficiency using relative and absolute metrics.

Data corresponding to the dataset's features are extracted using Table 3, and then both relative and absolute metrics are evaluated. Absolute metrics measure features and attributes for a set of data. They are employed to assess disparities between the attribute features of the training datasets and the generated music data. Relative metrics analyze the distributions of different dimensions. Two-by-two exhaustive cross-validation is primarily employed to calculate the distances between samples within or outside the datasets, resulting in distance histograms for each feature. Subsequently, kernel density is applied to smooth the distance histograms of features, facilitating the acquisition of the probability distribution function of the distance histogram for each feature.

As shown in Table 4, for the pitch count (PC), note count (NC), and inter-onset interval (IOI) features, the KLD inter-set distance between the Hyperbolic Music Transformer and our method, as well as the intra-set distance within our method, are smaller than the inter-set distance between the Euclidean Transformer and our method. In terms of OA (overall agreement) comparison analysis, the difference between the Hyperbolic Transformer and our method for inter-set and intra-set differences is higher than that of the Euclidean Transformer and our method's inter-set and intra-set distances. In the analysis of pitch range features, the Kullback-Leibler divergence inter-set and intra-set distances between the Hyperbolic Transformer and our approach exceed those between the

Table 3. The Feature Categories Used in the Objective Experiment Conducted With MGEVAL

Feature		Introduction
Pitch-based features	Pitch count (PC)	The count of distinct pitches within a sample
	Pitch range (PR)	The pitch range is computed 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

Table 4. OA and KLD Metrics Are Employed to Assess the Distribution of PC, PR, IOI, and NC Values Within Both the Generated Music and the Original Datasets, Providing a Means to Quantify Similarities

		Euclidean Transformer and our method	Hyperbolic Music Transformer and our method
PC	KLD	0.073	0.061
	OA	0.711	0.735
PR	KLD	0.038	0.059
	OA	0.780	0.864
IOI	KLD	0.355	0.108
	OA	0.768	0.833
NC	KLD	0.051	0.027
	OA	0.713	0.882

Euclidean Transformer and our method. However, from the OA analysis, the experimental results are the opposite, demonstrating that a direct comparison between the two is inconclusive. Therefore, we conclude that the music generated by our method aligns more closely with the standards, and compared with the Euclidean Transformer, our approach demonstrates superior proficiency in capturing the music features inherent in the original datasets.

Following Yang, we conducted a more in-depth objective comparison. First, we randomly selected 20 MIDI tracks from the music produced by Museformer, Euclidean Transformer, Hyperbolic Music Transformer, Longformer, and Linear Transformer, as well as our own method. We then analyzed these tracks using the MGEval tool, as shown in Table 5. The average values were computed for PR, NC, PC, and IOI. Notably, the music generated by our method exhibited metrics that were closer to those in the datasets, indicating richer generated music. This result demonstrates that our method is better at learning the style and characteristics of the dataset.

Perplexity (PPL). A widely used metric for assessing the predictive accuracy of a generative model is the repetition rate, which should ideally be minimized. To assess model performance across various text lengths, this metric is computed for the initial 1,024, 5,120, and 10,240 tokens in each sample.

Similarity Error (SE). To assess the capability of the models in generating music with authentic structures, we measure the discrepancy between the similarity distribution in the training data and the music produced by the model. This evaluation is quantified as shown in equation (2).

$$SE = \frac{1}{T}\sum_{t=1}^{T}\left|\hat{L}_t - L_t\right| \tag{2}$$

Table 5. Mean PC, PR, IOI, and NC Values Derived for the POP909 Datasets Based on Objective Assessments

	Musefor-mer	Euclidean Transformer	Hyperbolic Music Transformer	Longformer	Linear Transformer	Our method
PC	+0.11	+0.28	+0.31	-0.17	+0.16	-0.05
PR	-0.09	-0.33	-0.39	+0.15	+0.19	**+0.04**
IOI	-0.14	-0.31	+0.21	+0.18	-0.21	**-0.08**
NC	+0.18	-0.21	+0.29	-0.15	+0.19	**-0.10**

Note. When we used POP909 as a reference, smaller differences indicate a more effective emulation of the original dataset's style.

In this equation, $\hat{L}_t$ and L_t denote the mean similarities between the generated music and the training data, respectively. In our experiments, we designate T as 40, and we calculate C based on 100 pieces of generated music for each model. A lower value indicates a higher resemblance between the structures of the generated music and human-created music.

The objective evaluation results are presented in Table 6, revealing the following observations: Music Transformer demonstrates a similar perplexity to other models when applied to shorter music sequences (1,024 tokens). However, it experiences a substantial decline in performance when tasked with longer sequences. This finding suggests that a model trained on shorter music sequences struggles to generalize effectively to longer ones. Hence, there is a demand for an appropriate Transformer model designed to handle extended music sequences effectively. Despite the Linear Transformer's ability to cover the entire sequence through its receptive field, it does not exhibit superior PPL results compared with other models. This result could be ascribed to the kernel-based attention's incapacity to precisely capture the nuanced correlations present within the music. The newly introduced Museformer consistently outperforms other models in terms of PPL across various sequence lengths, particularly excelling on longer sequences. This finding underscores Museformer's effectiveness in the domain of music generation. The outcomes in terms of structural evaluation (SE) indicate that music generated by Museformer bears the closest resemblance to human-made music in terms of its structural characteristics.

Subjective Evaluation

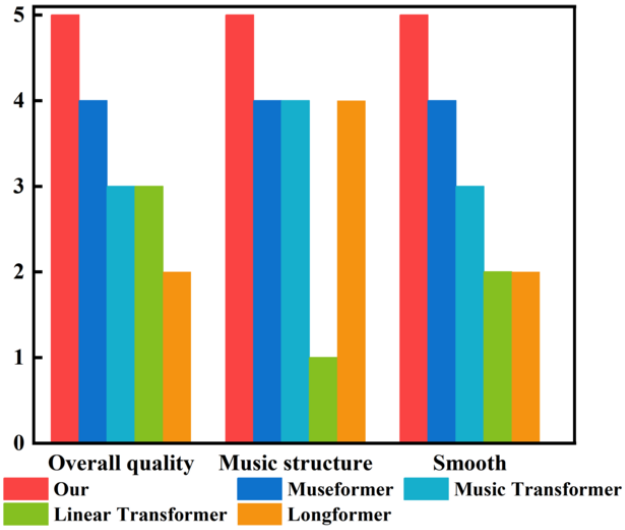
For the subjective evaluation experiment section, we randomly selected 20 pieces of music. Among them, 10 were from the music generated by our method, and the other 10 were from music in the POP909 datasets and other datasets. To ensure fairness, we randomly shuffled and evaluated these 20 pieces of music in a blind manner. We recruited a total of 20 participants, consisting of 10 males and 10 females. Note that five of them had a background in music, accounting for one-fourth of the total number.

In our experiments, we instructed participants to assign scores based on the following criteria: overall music quality, musical structure, and musical fluency. A rating of 1 represented the lowest score, whereas a rating of 5 denoted the highest score. As shown in Figure 3, our method achieved high scores of 5 in musical fluency, musical structure, and overall music quality, resulting in the highest overall score. Next, the Museformer method received a score of 4 in all three evaluation criteria, and the Music Transformer method scored lower overall than the first two methods. As described earlier, based on the assessments of the participants regarding the generated music, our method outperformed the other methods, demonstrating the effectiveness of our approach.

Table 6. Outcomes of Both Our Objective Evaluation and Ablation Study, With the Sequence Lengths Indicated Within Parentheses for the PPL Values

	SE (%)	PPL (1,024)	PPL (5,120)	PPL (10,240)
Longformer	5.25	1.65	1.46	1.45
Linear Transformer	1.97	1.86	1.67	1.64
Music Transformer	2.49	1.66	1.77	2.55
Museformer	0.95	1.64	1.41	1.35
Our method	**1.61**	**1.35**	**1.33**	**1.28**
- Start	1.58	1.29	1.31	1.55
- End	1.56	1.25	1.29	1.51

Figure 3. Experimenter's Evaluations Encompassing the Comprehensive Aspects of Music Quality, Structure, and Fluency



Describing the hierarchical structure of music can be quite challenging, so following Huang et al. (2018), we customized some relevant questions to specifically analyze the musical structure. This step allows for an accurate experimental analysis of our method, as shown in Table 7. Regarding the experimental setup, it remained consistent with the previous setup, where the options for the experiment were set to yes or no." Consequently, the number of participants who chose yes for our method follows the pattern as indicated in Table 7. Our method had the highest preference, followed by Museformer and Euclidean Transformer, with Hyperbolic Music Transformer having the lowest preference. The final question pertained to the presence of noisy notes in the music, indicating that the more participants chose yes, the fewer noisy notes there were, making the overall music more harmonious. Table 7 also indicates that our method produced the fewest noisy notes. In summary, the music generated by our method demonstrated a significant advantage in the aforementioned evaluation criteria.

Table 7. Experimental Statistical Outcomes Regarding the Music's Structural Configurations are Presented in Percentage Format

	Museformer	Euclidean Transformer	Hyperbolic Music Transformer	Our
Did you feel phrase?	89.1%	88.4%	87.6%	92.2%
Smooth transitions between phrase?	90.2%	89.6%	85.1%	93.1%
Does the pitch change harmoniously?	91.4%	84.6%	82.3%	94.6%
Is the rhythm comfortable?	89.9%	80.5%	85.1%	93.8%
Does music have a hierarchy?	81.7%	78.8%	77.5%	90.9%
Music overall harmony?	91.2%	87.1%	85.6%	93.3%
Are there abrupt notes?	40.3%	42.2%	47.1%	36.4%

Compared with alternative models (comparative modeling and ablation configurations), our generated music exhibits the closest resemblance to the training data's similarity distribution, as evidenced by its smallest SE value as shown in Table 6. Nevertheless, it's worth noting that SE may not fully capture all structural characteristics.

In Figure 4, the similarity distributions are presented, revealing the following observations: First, Museformer's distribution bears a striking resemblance to that of the training data, with closely aligned quantities and a contour displaying an identical periodic pattern.

Second, both the Music Transformer and Linear Transformer distributions lack a discernible periodic pattern. This finding suggests that, when trained on shorter sequences, the Music Transformer model struggles to generate well-structured music of extended lengths, while the Linear Transformer fails to adequately capture the structure-related correlations, despite its expansive receptive field spanning the entire sequence.

Third, the distributions observed in Transformer-XL and Longformer exhibit a propensity for a periodic pattern, with a general decline in similarity as the interval increases. This finding suggests that these models, characterized by receptive fields encompassing primarily the most recent content, can generate periodic repetitions over short distances, but struggle to capture long-term structural patterns effectively.

Fourth, apparently in human evaluations of structure-related metrics, the contour (i.e., the periodic pattern) carries greater significance compared with the quantity of similarity. Transformer-XL and Longformer exhibit this pattern in their distributions, resulting in relatively high subjective scores for both short- and long-term structures, surpassing those of Music Transformer and Linear Transformer, both of which lack the periodic pattern. Nevertheless, it's important to acknowledge that similarity quantity can also impact human assessments. For instance, Transformer-XL often yields relatively high similarity values, indicating an excess of repetitions in some instances. In such cases, human evaluators may find these repetitions irritating, ultimately resulting in lower scores for musicality.

Fifth, in the ablation setting, Museformer excluding coarse-grained attention shows a marginally greater structural error compared with the standard Museformer. However, its distribution distinctly reveals a periodic pattern. Consequently, it appears that the coarse-grained attention module makes only a marginal contribution to the overall music structures. Conversely, the distribution of Museformer without bar selection displays a tendency toward the periodic pattern, with a general decrease in similarity as the interval increases, akin to the patterns observed in Transformer-XL and Longformer. This finding implies that incorporating bars related to structure is crucial for producing music with both short- and long-term structural characteristics.

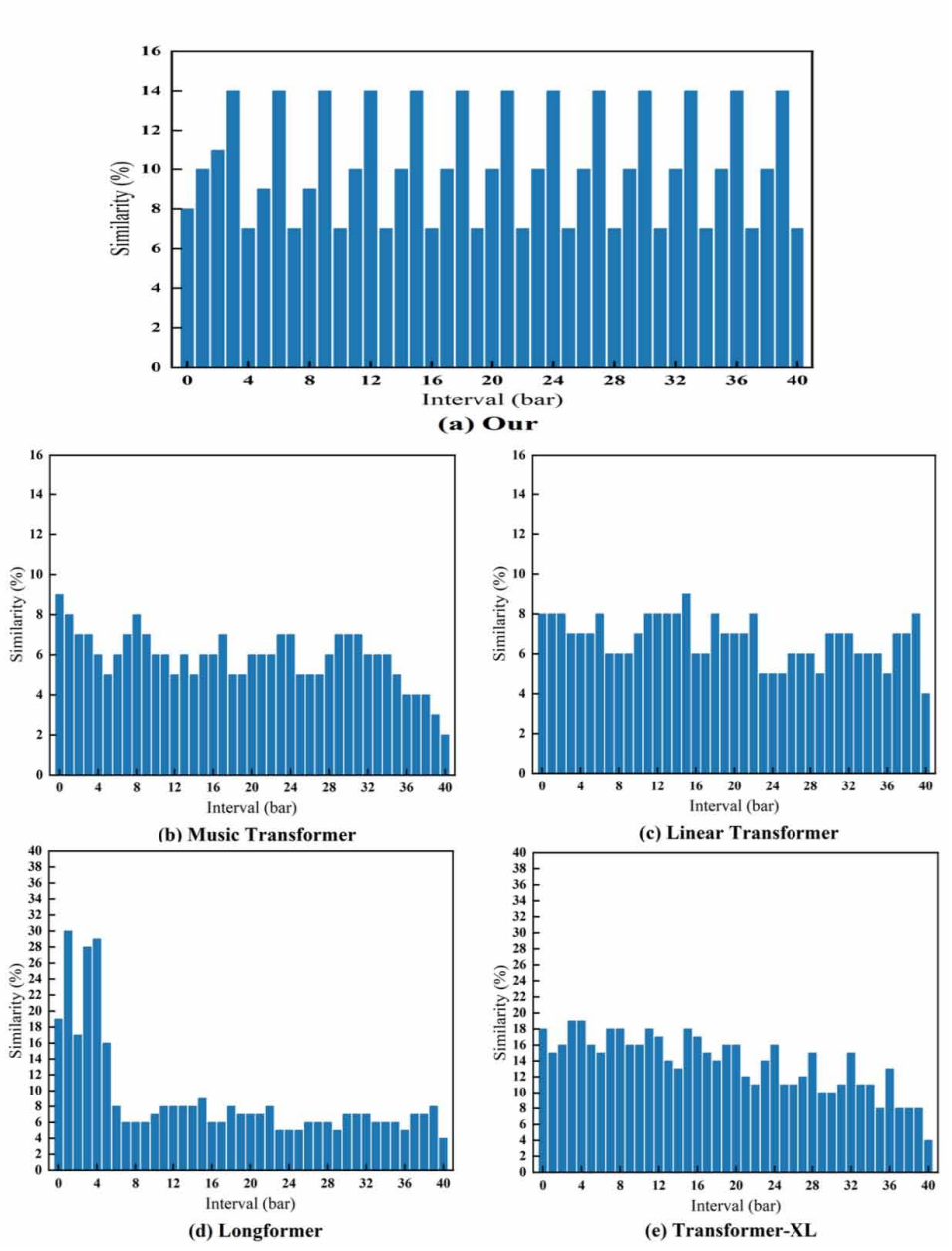
Ablation Study

Upon the data displayed in Table 6, the following observations emerge: First, our method consistently outperforms both ablation settings in terms of PPL and SE, indicating the effectiveness of including Start and End tokens in the settings. Second, the Start token is effective in initializing essential music-related information related to structure. Third, as the sequence length increases, the contribution of music structure-related selections becomes more significant. This rationale stems from the fact that longer music sequences typically encompass an increased number of measures and exhibit more extensive structural intricacies. The End token can directly capture the relevance of measures related to distant structures, which helps in making more accurate endings. Fourth, the selection of Start and End tokens contributes to the generation of the music's structure, particularly in terms of SE.

CONCLUSION

In this paper, we introduced a novel Sequence-to-Music Transformer music generation model. The goal of our work was to generate music with strong coherence, hierarchical structure, and smooth transitions. Leveraging Transformer-based music generation methods, our approach encoded the

Figure 4. Distribution of Similarity in the Melody Track of Music Generated by Various Models



musical content in the encoder to express the hierarchical structure of music. The proposed decoder structure captures the hierarchical dependencies in music, which in turn helps generate music with better hierarchical structures. Comprehensive experiments confirmed the efficacy of our approach in producing high-quality music quality, further corroborated by objective validation.

Although our approach achieved reasonably good performance in music generation, dealing with the case of composing for multiple voices or even for an entire symphony was difficult. Moreover, we dealt with paired phrases with echo patterns in the melody. The reason for this may be that we

modeled our framework with only a relatively small number of music fragments. A simpler way to handle this would be to input the entire music to model it and train the model to output the music fragment by fragment in an auto-regressive manner. In future work, our team will study and model such longer music fragments in preparation for producing higher quality music.

ABBREVIATIONS AND TECHNICAL TERMS

Hidden Markov Models (HMM)
Variable Markov Oracle (VMO)
Information Rate (IR)
Pitch Count (PC)
Pitch Range (PR)
Note Count (NC)
Inter-onset Interval (IOI)
Overall Agreement (OA)
Perplexity (PPL)
Similarity Error (SE)

ACKNOWLEDGMENT

All work was conducted while the authors were affiliated with the Shangqiu Normal University, which was prior to other affiliations.

We confirm that informed consent was obtained from all subjects and/or their legal guardian(s).

AUTHOR CONTRIBUTIONS

Conceptualization, Y.X.; methodology, Y.X.; software, Y.X.; validation, Y.X.; formal analysis, Y.X.; investigation, Y.X.; resources, Y.X.; data curation, Y.X.; writing—original draft preparation, Y.X.; writing—review and editing, Y.X.; visualization, Y.X.; supervision, Y.X.; project administration, Y.X.; funding acquisition, Y.X. All authors have read and agreed to the published version of the manuscript.

CONFLICTS OF INTEREST

We wish to confirm that there are no known conflicts of interest associated with this publication and there has been no significant financial support for this work that could have influenced its outcome.

FUNDING STATEMENT

No funding was received for this work.

PROCESS DATES

Received: 2/28/2024, Revision: 3/20/2024, Accepted: 3/20/2024

REFERENCES

Babalola, O. P., Usman, A. M., Ogundile, O. O., & Versfeld, D. J. J. (2021). Detection of Bryde's whale short pulse calls using time domain features with hidden Markov models. *SAIEE Africa Research Journal.*, *112*(1), 15–23. doi:10.23919/SAIEE.2021.9340533

Barbosa, A., Bittencourt, I. I., Siqueira, S. W., Dermeval, D., & Cruz, N. J. T. (2022). A context-independent ontological linked data alignment approach to instance matching. [IJSWIS]. *International Journal on Semantic Web and Information Systems*, *18*(1), 1–29. doi:10.4018/IJSWIS.295977

Bronstein, M. M., Bruna, J., LeCun, Y., Szlam, A., & Vandergheynst, P. (2017). Geometric deep learning: Going beyond Euclidean data. *IEEE Signal Processing Magazine*, *34*(4), 18–42. doi:10.1109/MSP.2017.2693418

Chen, X., Li, J., & Zhang, Y. F. (2022). Multidirectional gradient feature with shape index for effective texture classification. [IJSWIS]. *International Journal on Semantic Web and Information Systems*, *18*(1), 1–19. doi:10.4018/IJSWIS.312183

Choi, K., Hawthorne, C., Simon, I., Dinculescu, M., & Engel, J. (2020). Encoding musical style with transformer autoencoders. In *Proceedings of the 37th International Conference on Machine Learning, 119*, 1899–1908. MLResearch Press. https://proceedings.mlr.press/v119/choi20b.html

Chopra, M., Singh, S. K., Sharma, A., & Gill, S. S. (2022). A comparative study of generative adversarial networks for text-to-image synthesis. [IJSSCI]. *International Journal of Software Science and Computational Intelligence*, *14*(1), 1–12. doi:10.4018/IJSSCI.300364

Dai, S., Jin, Z., Gomes, C., & Dannenberg, R. B. (2021). Controllable deep melody generation via hierarchical music structure representation. In *Proceedings of the 22nd International Society for Music Information Retrieval Conference*, 143–150. arXiv:2109.00663 [cs.SD]. doi:10.48550/arXiv.2109.00663

Deutsch, D., & Feroe, J. (1981). The internal representation of pitch sequences in tonal music. *Psychological Review*, *88*(6), 503–522. https://psycnet.apa.org/doi/10.1037/0033-295X.88.6.503. doi:10.1037/0033-295X.88.6.503

Huang, W., Yu, Y., Xu, H., Su, Z., & Wu, Y. (2023). Hyperbolic music transformer for structured music generation. *IEEE Access : Practical Innovations, Open Solutions*, *11*, 26893–26905. doi:10.1109/ACCESS.2023.3257381

Ilyas, Q. M., Ahmad, M., Rauf, S., & Irfan, D. (2022). RDF query path optimization using hybrid genetic algorithms: Semantic web vs. data-intensive cloud computing. [IJCAC]. *International Journal of Cloud Applications and Computing*, *12*(1), 1–16. doi:10.4018/IJCAC.2022010101

Ismail, S., Shishtawy, T. E., & Alsammak, A. K. (2022). A new alignment word-space approach for measuring semantic similarity for Arabic text. [IJSWIS]. *International Journal on Semantic Web and Information Systems*, *18*(1), 1–18. doi:10.4018/IJSWIS.297036

Jiang, J., Xia, G. G., Carlton, D. B., Anderson, C. N., & Miyakawa, R. H. (2020). Transformer VAE: A hierarchical model for structure-aware and interpretable music representation learning. In *Proceedings of the 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2020), pp.* 516–520. IEEE. doi:10.1109/ICASSP40776.2020.9054554

Krauss, O. (2023). Exploring the use of natural language processing techniques for enhancing genetic improvement. In *Proceedings of the IEEE/ACM International Workshop on Genetic Improvement (GI)*, pp. 21–22. IEEE. doi:10.1109/GI59320.2023.00014

Lee, J. M. (2003). *Introduction to smooth manifolds* (2nd ed.). Springer., doi:10.1007/978-0-387-21752-9

Lee, J. M. (2018). *Introduction to Riemannian manifolds* (2nd ed.). Springer., doi:10.1007/978-3-319-91755-9

Lerdahl, F. (2001). Tonal pitch space. *Music Perception*, *5*(3), 315–349. doi:10.2307/40285402

Lerdahl, F., & Jackendoff, R. (1983). An overview of hierarchical structure in music. *Music Perception*, *1*(2), 229–252. doi:10.2307/40285257

Li, D., Deng, L., Gupta, B. B., Wang, H., & Choi, C. (2019). A novel CNN based security guaranteed image watermarking generation scenario for smart city applications. *Information Sciences*, *479*, 432–447. doi:10.1016/j.ins.2018.02.060

Li, M., Miao, Z., Zhang, X.-P., Xu, W., Ma, C., & Xie, N. (2022). Rhythm-aware sequence-to-sequence learning for labanotation generation with gesture-sensitive graph convolutional encoding. *IEEE Transactions on Multimedia*, *24*, 1488–1502. doi:10.1109/TMM.2021.3066115

Lin, T.-Y., Goyal, P., Girshick, R., He, K., & Dollar, P. (2020). Focal loss for dense object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *42*(2), 318–327. doi:10.1109/TPAMI.2018.2858826 PMID:30040631

Liu, D., Hong, Y., Yao, J., & Zhou, G. (June 2023). Question generation via generative adversarial networks. In *Proceedings of the 2023 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8. IEEE. doi:10.1109/IJCNN54540.2023.10191871

Liu, L., Qian, L., Zhang, Q., Ding, L., Yang, F., & Shi, Y. (April 2023). Optimized embedded model for time sequence alignment within distributed video streams environment. In *Proceedings of 2023 3rd International Conference on Information Communication and Software Engineering (ICICSE)*, pp. 9–14. IEEE. doi:10.1109/ICICSE58435.2023.10211942

Liu, R. W., Guo, Y., Lu, Y., Chui, K. T., & Gupta, B. B. (2022, April). Deep network-enabled haze visibility enhancement for visual IoT-driven intelligent transportation systems. *IEEE Transactions on Industrial Informatics*, *19*(2), 1581–1591. doi:10.1109/TII.2022.3170594

Ma, L., Zhao, Y., Wang, B., & Shen, F. (2023). A multistep sequence-to-sequence model with attention LSTM neural networks for industrial soft sensor application. *IEEE Sensors Journal*, *23*(10), 10801–10813. doi:10.1109/JSEN.2023.3266104

Memos, V. A., Psannis, K. E., Ishibashi, Y., Kim, B.-G., & Gupta, B. B. (2018). An efficient algorithm for media-based surveillance system (EAMSuS) in IoT smart city framework. *Future Generation Computer Systems*, *83*, 619–628. doi:10.1016/j.future.2017.04.039

Moysis, L., Lliadis, L. A., Sotiroudis, S. P., Boursiansis, A. D., Papadopoulou, M. S., Kokkinidis, K., Volos, C., Sarigiannidis, P., Nikolaidis, S., & Goudos, S. K. (2023, February). Music deep learning: Deep learning methods for music signal processing—A review of the state-of-the-art. *IEEE Access : Practical Innovations, Open Solutions*, *11*, 17031–17052. doi:10.1109/ACCESS.2023.3244620

Moysis, L., Lliadis, L. A., Sotiroudis, S. P., Kokkinidis, K., Sarigiannidis, P., Nikolaidis, S., Volos, C., Boursiansis, A. D., Babas, D., Papadopoulou, M. S., & Goudos, S. K. (July 2023). The challenges of music deep learning for traditional music. In *Proceedings of the 2023 12th International Conference on Modern Circuits and Systems Technologies (MOCAST)*, 1–5. IEEE. doi:10.1109/MOCAST57943.2023.10176775

Mushtaq, U., & Cabessa, J. (2023). Argument mining with modular BERT and transfer learning. In *Proceedings of the 2023 International Joint Conference on Neural Networks (IJCNN)*, 1–8. IEEE. doi:10.1109/IJCNN54540.2023.10191968

Nguyen, G. N., Le Viet, N. H., Elhoseny, M., Shankar, K., Gupta, B. B., & El-Latif, A. A. A. (2021). Secure blockchain enabled cyber-physical systems in healthcare using deep belief network with ResNet model. *Journal of Parallel and Distributed Computing*, *153*, 150–160. doi:10.1016/j.jpdc.2021.03.011

Papadopoulos, G., & Wiggins, G. A. (1999). AI methods for algorithmic composition: A survey, a critical view, and future prospects. *In Proceedings of the AISB'99 Symposium on Musical Creativity*, pp. 110–117. AISB (Society for the Study of Artificial Intelligence and the Simulation of Behaviour).

Pei, Y., Zhao, J., Yao, Y., & Ding, F. (2023). Multi-task reinforcement learning for distribution system voltage control with topology changes. *IEEE Transactions on Smart Grid*, *14*(3), 2481–2484. doi:10.1109/TSG.2022.3233766

Peng, W., Varanka, T., Mostafa, A., Shi, H., & Zhao, G. (2022). Hyperbolic deep neural networks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *44*(12), 10023–10044. doi:10.1109/TPAMI.2021.3136921 PMID:34932472

Raffel, C. (2016). Learning-based methods for comparing sequences, with applications to audio-to-MIDI alignment and matching [Doctoral dissertation, Columbia University]. 10.7916/D8N58MHV

Sarivougioukas, J., & Vagelatos, A. (2022). Fused contextual data with threading technology to accelerate processing in home UbiHealth. [IJSSCI]. *International Journal of Software Science and Computational Intelligence*, *14*(1), 1–14. doi:10.4018/IJSSCI.285590

Shen, W. (2023). Design of video background music generation algorithm based on music feature matching algorithm. In *Proceedings of the 2023 Asia-Europe Conference on Electronics, Data Processing and Informatics (ACEDPI)*, 122–126. IEEE. doi:10.1109/ACEDPI58926.2023.00032

Singh, S. K., & Sachan, M. K. (2021). Classification of code-mixed bilingual phonetic text using sentiment analysis. [IJSWIS]. *International Journal on Semantic Web and Information Systems*, *17*(2), 59–78. doi:10.4018/IJSWIS.2021040104

Wang, Z., Chen, K., Jiang, J., Zhang, Y., Xu, M., Dai, S., Gu, X., & Xia, G. (2020). POP909: A pop-song dataset for music arrangement generation. In *Proceedings of the 21st International Society for Music Information Retrieval Conference*. arXiv:2008.07142 [cs.SD]. doi:10.48550/arXiv.2008.07142

Wu, Z., Tran, H., Pirsiavash, H., & Kolouri, S. (2023). Is multi-task learning an upper bound for continual learning? In *Proceedings of the 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2023)*, pp. 1–5. IEEE. doi:10.1109/ICASSP49357.2023.10095984

Yen, S., Moh, M., & Moh, T.-S. (2021). Detecting compromised social network accounts using deep learning for behavior and text analyses. [IJCAC]. *International Journal of Cloud Applications and Computing*, *11*(2), 97–109. doi:10.4018/IJCAC.2021040106

Yu, B., Lu, P., Wang, R., Hu, W., Tan, X., Ye, W., Zhang, S., Qin, T., & Liu, T.-Y. (2021). Museformer: Transformer with fine- and coarse-grained attention for music generation. In *Proceedings of the 36th Conference on Neural Information Processing Systems (NeurIPS 2022)*, pp. 1–20. arXiv:2210.10349 [cs.SD]. doi:10.48550/arXiv.2210.10349

Zhang, N. (2023). Learning adversarial transformer for symbolic music generation. *IEEE Transactions on Neural Networks and Learning Systems*, *34*(4), 1754–1763. doi:10.1109/TNNLS.2020.2990746 PMID:32614773

Zhang, Q., Guo, Z., Zhu, Y., Vijayakumar, P., Castiglione, A., & Gupta, B. B. (2023). A deep learning-based fast fake news detection model for cyber-physical social services. *Pattern Recognition Letters*, *168*, 31–38. doi:10.1016/j.patrec.2023.02.026

Zheng, N., Lin, B., Zhang, Q., Ma, L., Yang, Y., Yang, F., Wang, Y., Yang, M., & Zhou, L. (2022). Sparta: Deep-learning model sparsity via tensor-with-sparsity-attribute. In *Proceedings of the 16th USENIX Symposium on Operating Systems Design and Implementation (OSDI)*, 213–232. USENIX Association. https://www.usenix.org/system/files/osdi22-zheng-ningxin.pdf

Zhou, Y., Yu, T., Gao, W., Huang, W., Lu, Z., Huang, Q., & Li, Y. (2023). Shared three-dimensional robotic arm control based on asynchronous BCI and computer vision. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, *31*, 3163–3175. doi:10.1109/TNSRE.2023.3299350 PMID:37498753

Yang Xu, Lecturer, Doctorate, Graduated from Korea Kookmin University in 2021, and worked in Shangqiu Normal University. His research interests include Art Studies and Artificial Intelligence.