GENPIA: A Genre-Conditioned Piano Music Generation System

Quoc-Viet Nguyen*, Hao-Wei Lai*, Khanh-Duy Nguyen*, Min-Te Sun*, Wu-Yuin Hwang†,Kazuya Sakai‡, Wei-Shinn Ku§
* Computer Science and Information Engineering, National Central University, Taoyuan, Taiwan
† Graduate Institute of Network Learning Technology, National Central University, Taoyuan, Taiwan
‡ Electrical Engineering and Computer Science, Tokyo Metropolitan University, Hino, Tokyo, Japan
§ Computer Science and Software Engineering, Auburn University, Auburn, AL, USA
Email: vietnq@uit.edu.vn, yellow946821@gmail.com, nkduy@tvu.edu.vn, msun@csie.ncu.edu.tw
wyhwang1206@gmail.com, ksakai@tmu.ac.jp, weishinn@auburn.edu

***Abstract*—With the demand for music continuing to grow as people seek variety and personal resonance, many works focus on music generation. In this study, we propose GENPIA, a genre-conditioned piano music generation system. The system encompasses Anime, R&B, Jazz, and Classical music genres. To build our system, we collect and label audio data of various genres for the specific objective of our research. REMI audio representation with genre information extension is applied during data pre-processing to present the audio data with a better data structure. Transformer-XL is implemented as the model to learn knowledge about the extended audio representation and generate the desired output audio. An external dataset, called Ailabs.tw 1K7, is utilized for pre-training purposes. The results obtained from a listening questionnaire show that GENPIA can generate better piano pieces conditioned on different genres compared to the prior state-of-the-art work.**

***Index Terms*—GENPIA, Piano music generation, Genre-condition, Transfromer-XL.**

I. Introduction

Generative AI has attracted increasing attention alongside advancements in deep learning technologies. For instance, innovations like ChatGPT proposed by OpenAI [33] and Midjourney' [31] text-to-image generation capabilities exemplify the progress in text-to-text and text-to-image generation, respectively. Despite these achievements [2, 39], generating music from textual input remains a challenging endeavor with ongoing efforts striving for improvement, especially concerning diverse music genres.

To address this challenge, leveraging deep learning for music generation has emerged as a promising approach, drawing significant interest due to the intrinsic connection between music and deep learning. Initially, methods based on recurrent neural networks (RNNs) have been employed for [37] for music generation tasks. However, the introduction of Transformer [43] architectures has revolutionized the field, with many researchers adopting Transformer-based approaches. Nevertheless, variations persist among these methods, including differences in data pre-processing, data types, and the genres of music generated.

Music composition demands extensive musical knowledge, posing significant challenges even for human composers. In the genre-conditioned piano music generation research, the three main challenges are:

1) The structure of music: Improving musical structure can increase the listener's ability to understand and appreciate the music. When writing articles, we utilize different paragraphs to separate various information we intend to describe, which increases the reader's comprehension. To improve audio generation capabilities, enhancing the structure of the music is necessary.
2) Genre information addition: To generate piano music that is conditioned on genre, it is crucial and essential to incorporate the genre label information into the audio data during the pre-processing stage. In addition, the method of genre information addition should support model inference.
3) Conditioned by genre: In Section II-B, we emphasize that EMOPIA [22] demonstrates the capacity to capture emotional cues in music, yielding exceptional results across various assessments. However, it remains uncertain whether similar success can be attained when shifting focus from emotions to genres.

Taking inspiration from EMOPIA [22], which demonstrates the effectiveness of Transformer-based models in emotion-conditioned piano music generation and addresses key research agendas in the scientific field of the Internet of Sounds [42] our study shifts its focus to genre-conditioned piano music generation. Specifically, we explore genres such as Anime, R & B, Jazz, and Classical. In summary, the contributions of this research can be summarized as follows:

- We developed a system called GENPIA that is able to automatically generate piano music based on a specified target music genre.
- We gathered and meticulously labeled a music dataset, conducting thorough data cleansing to meet the requirements of REMI [21] audio representation. In addition, we implemented genre information addition to enhance the data structure for representing audio data, converting it into REMI audio representation with extension.
- To learn audio patterns across various music genres,

we adopt Transformer-XL [9], which has a high input dependency length, and we utilize the proposed method of genre information addition to support model inference.

- In addition to utilizing our dataset for model training, we incorporate an external dataset, Ailabs.tw 1K7 [19], for pre-training purposes.
- We designed a listening questionnaire incorporating subjective metrics to conduct a comparative survey between GENPIA and EMOPIA [22], with and without pre-training, across various music genres. The survey results demonstrate GENPIA's superior ability to generate piano music across different genres.

II. Related work

A. Non-Transformer-based Music Generation

Most of the works that are not Transformer-based on music generation are based on Long Short-Term Memory (LSTM) [18]. In [23], a model based on LSTM and Reinforcement Learning (RL) [25], is trained on a huge library of songs. This method not only generates more attractive melodies but also significantly reduces the occurrence of errors in Recurrent Neural Networks (RNNs) [37]. Wu *et al.* [46] merge three LSTM subnetworks to present a hierarchical RNN that performs better than a single one. Roberts *et al.* [36] construct a Variational Autoencoder (VAE) [28] with LSTM model and make use of a hierarchical decoder to deal with longer music generations. Apart from LSTM models, Yang *et al.* [49] apply Convolutional Neural Networks (CNNs) [48] and transform them into a Generative Adversarial Network (GAN) [17] for composing music with multiple MIDI tracks.

B. Transformer-based Music Generation

The earlier Transformer-based methods [8, 20, 24] use MIDI-like audio representation [32] , and their models are trained on the Transformer introduced in [43]. Huang *et al.* [20] apply the relative attention mechanism to compose minute-long music. Choi *et al.* [8] combine raw data encodings periodically to create an overall representation. Jiang *et al.* [24] incorporate the works of learning understandable latent representations and relationships over time. The works in [12, 35] improve the MIDI-like audio representation with extensions. Ens *et al.* [12] convert multiple soundtracks into a single sequence, each with a time-ordered series of musical events, and finally name their model MMM. After the introduction of Transformer-XL [9], Donahue *et al.* [11] use MIDI-like audio representation with extensions. This combination improves multi-instrumental music generation. Then, Chen and Wu [6, 45] apply REMI with extensions and generate music using Transformer-XL. Chen *et al.* [6] propose fingerstyle guitar tabs while Wu *et al.* [45] focus on lead sheets of Jazz music. Last but not least, Hung *et al.* [22] make use of Compound Word [19] on Linear Transformer [26] to perform emotion-conditioned music generation. The main advantage of Linear Transformer is its low time and memory complexity during training and inference.

C. Music Datasets

Music datasets can be categorized into labeled and unlabeled datasets:

1) **Unlabeled music datasets**: Defferrard *et al.* [10] provide Free Music Archive (FMA) dataset, which includes audio and metadata (e.g., duration, license, and producer). Bittner *et al.* [5] introduce MedleyDB a dataset comprising audio, annotations (e.g., source id, pitch, and melody), and metadata (e.g., artist, composer, and producer). Kong *et al.* [29] created GiantMIDI, a large MIDI dataset of classical piano music. Wang *et al.* [44] present POP909, which contains 909 popular songs with piano accompaniment, lead instrument melody, and vocal melody in MIDI format. Hsiao *et al.* [19] also collected a dataset with pop piano performance, called Ailabs.tw 1K7.
2) **Labeled music datasets**: The most commonly used label type is emotion-related. For instance, the datasets in [7, 34] use adjectives (e.g., happy, inspiring, dark, and tense) as labels. The works in [13, 14] label the music with valence and arousal values, especially in [22], which utilizes Russell's 4Q [38] (i.e., high/low of valence and high/low of arousal) as its label type.

III. Preliminary

A. Transformer-XL

Transformer-XL [9] model uses the architecture based on Transformer [43] shown in Figure 4. In the following subsections, we delve into the specific architectures of the encoder and decoder employed in Transformer-XL. Furthermore, we explain the improvements of Transformer-XL which distinguish it from the foundational Transformer model.

1) Transformer Encoder: The encoder (in Figure 1) starts with multi-head attention [43], then adds the original inputs using a residual connection and normalizes them. Afterward, it passes through Layer Normalization [4]. Next, the outputs go through a fully connected network with another residual connection. Finally, Layer Normalization is applied again.

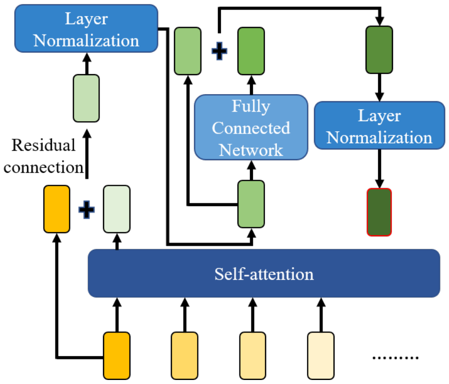


Fig. 1: The encoder processes in Transformer-XL.

2) Transformer Decoder: Contrary to the encoder, the decoder (Figure 2) starts with an initial token. It then uses previous outputs as inputs, determining its outputs through a probability distribution. The key difference between encoder and decoder lies in masked multi-head attention, essential because the decoder accesses previous outputs incrementally. Hence, masked multi-head attention is employed to overcome this limitation. Figure 3 illustrates this distinction between masked and non-masked multi-head attention [43]. As shown in Figure 4, Transformer-XL's encoder employs cross-attention to connect with the decoder.

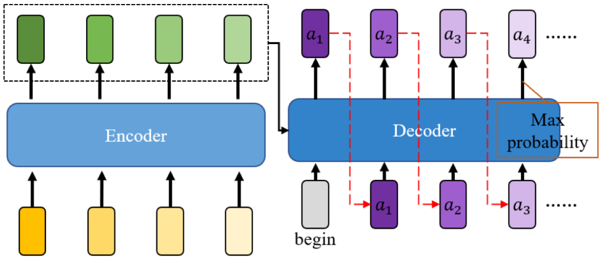
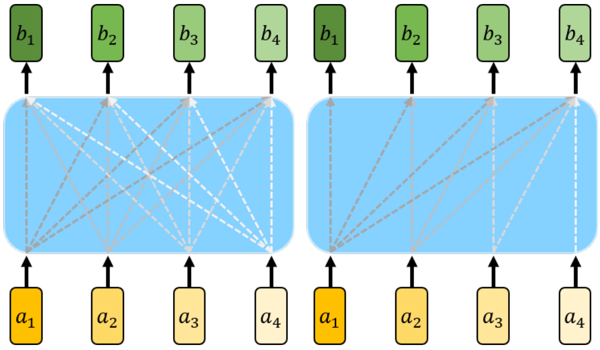


Fig. 2: The decoder processes in Transformer-XL.



(a) Multi-head Attention. (b) Masked Multi-head Attention.

Fig. 3: The comparison of Multi-Head Attention with and without mask.

3) Differences between Transformer and Transformer-XL: Transformer and Transformer-XL differ in Segment-Level Recurrence, Attention with Relative Positional Encoding, and Stochastic Temperature-Controlled Sampling, discussed in subsequent subsections.

a) Segment-Level Recurrence: Transformer's fixed input length limits its ability to model dependencies across longer segments. Transformer-XL overcomes this with segment-level recurrence, connecting segments and forecasting future tokens. In Figure 5, Transformer-XL's attention mechanisms are depicted during training and inference with a fixed segment length of 4. In Figure 5(a), previous segment embeddings are cached for reuse, shown by green lines. In Figure 5(b), Transformer-XL utilizes information not just within the current segment but also from previous ones, depicted by the green area. Segment-level recurrence significantly extends dependency length, addressing the context fragmentation problem [3, 9].

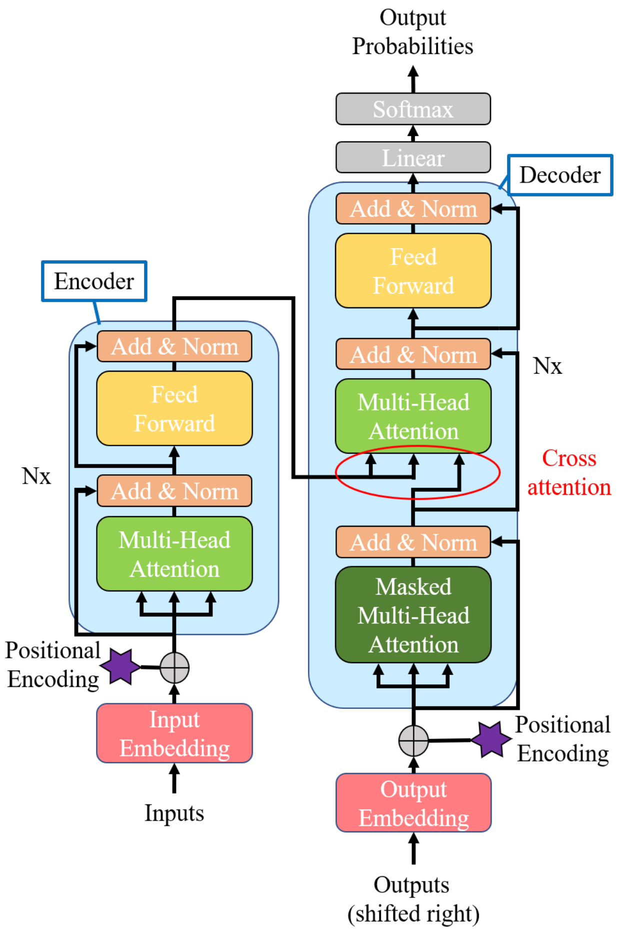


Fig. 4: The architecture of Transformer.

b) Attention Using Relative Positional Encoding: To apply segment-level recurrence, positional encoding may prove unsuitable due to position incoherence. The attention result between positions i and j using positional encoding [43], denoted as $\mathbf{A}_{i,j}^{abs}$ is described by Equation 1. Here $\mathbf{E}_{x_i}$ denotes the content embedding of x_i, $\mathbf{W}_q$ and $\mathbf{W}_k$ denote the weight matrices of query and key vectors, respectively, and $\mathbf{U}_i$ signifies the positional encoding of position i. Equation (2a) describes the attention between the content embedding added with positional encoding of positions i and j. Expanding Equation (1a) to Equation (1b) breaks it into four parts, each defining the attention between different elements. Specifically, part (a) represents the attention between $\mathbf{E}_{x_i}$ and $\mathbf{E}_{x_j}$, part (b) between $\mathbf{E}_{x_i}$ and $\mathbf{U}_j$, part (c) between $\mathbf{U}_i$ and $\mathbf{E}_{x_j}$, and part (d) between $\mathbf{U}_i$ and $\mathbf{U}_j$.

Modified from Positional Encoding, Dai *et al.* [9] introduce the attention result of relative positional encoding as shown in Equation 2. This equation can be divided into four parts, each corresponding to the symbol characters marked in Equation (1b). In Equation (2a), the $\mathbf{U}_j$ term in Equation (1a) is replaced with $\mathbf{R}_{i-j}$, a sinusoid encoding matrix [43] that utilizes the positional difference between i and j to denote relative position. Upon expanding Equation (2a) to Equation (2b), we can employ the same conceptual framework as Equation (1b) to comprehend it. Recognizing that utilizing the same weight matrix $\mathbf{W}_k$ to compute both content and positional information may be suboptimal, the authors introduce two distinct weight matrices for the key vector: $\mathbf{W}_{k,E}$ and $\mathbf{W}_{k,R}$, replacing $\mathbf{W}_k$ in Equation (2b) for the computation of content and positional information, respectively. However, given that the query vector remains constant for all query positions, the attention towards different words should remain consistent. Consequently, the authors introduce two learnable parameters, $u^\top$ and $v^\top$, to individually replace the term $\mathbf{U}_i^\top \mathbf{W}_q^\top$ of Equation (2c)in parts (c) and (d) to obtain Equation (2d).

$$\mathbf{A}_{i,j}^{abs} = (\mathbf{E}_{x_i} + \mathbf{U}_i)^\top \mathbf{W}_q^\top \mathbf{W}_k (\mathbf{E}_{x_j} + \mathbf{U}_j) \tag{1a}$$

$$\begin{aligned} &= \underbrace{\mathbf{E}_{x_i}^\top \mathbf{W}_q^\top \mathbf{W}_k \mathbf{E}_{x_j}}_{\text{(a)}} + \underbrace{\mathbf{E}_{x_i}^\top \mathbf{W}_q^\top \mathbf{W}_k \mathbf{U}_j}_{\text{(b)}} \\ &+ \underbrace{\mathbf{U}_i^\top \mathbf{W}_q^\top \mathbf{W}_k \mathbf{E}_{x_j}}_{\text{(c)}} + \underbrace{\mathbf{U}_i^\top \mathbf{W}_q^\top \mathbf{W}_k \mathbf{U}_j}_{\text{(d)}} \end{aligned} \tag{1b}$$

$$\mathbf{A}_{i,j}^{rel} = (\mathbf{E}_{x_i} + \mathbf{U}_i)^\top \mathbf{W}_q^\top \mathbf{W}_k (\mathbf{E}_{x_j} + \mathbf{R}_{i-j}) \tag{2a}$$

$$\begin{aligned} &= \underbrace{\mathbf{E}_{x_i}^\top \mathbf{W}_q^\top \mathbf{W}_k \mathbf{E}_{x_j}}_{\text{(a)}} + \underbrace{\mathbf{E}_{x_i}^\top \mathbf{W}_q^\top \mathbf{W}_k \mathbf{R}_{i-j}}_{\text{(b)}} \\ &+ \underbrace{\mathbf{U}_i^\top \mathbf{W}_q^\top \mathbf{W}_k \mathbf{E}_{x_j}}_{\text{(c)}} + \underbrace{\mathbf{U}_i^\top \mathbf{W}_q^\top \mathbf{W}_k \mathbf{R}_{i-j}}_{\text{(d)}} \end{aligned} \tag{2b}$$

$$\begin{aligned} &\Rightarrow \underbrace{\mathbf{E}_{x_i}^\top \mathbf{W}_q^\top \mathbf{W}_{k,E} \mathbf{E}_{x_j}}_{\text{(a)}} + \underbrace{\mathbf{E}_{x_i}^\top \mathbf{W}_q^\top \mathbf{W}_{k,R} \mathbf{R}_{i-j}}_{\text{(b)}} \\ &+ \underbrace{\mathbf{U}_i^\top \mathbf{W}_q^\top \mathbf{W}_{k,E} \mathbf{E}_{x_j}}_{\text{(c)}} + \underbrace{\mathbf{U}_i^\top \mathbf{W}_q^\top \mathbf{W}_{k,R} \mathbf{R}_{i-j}}_{\text{(d)}} \end{aligned} \tag{2c}$$

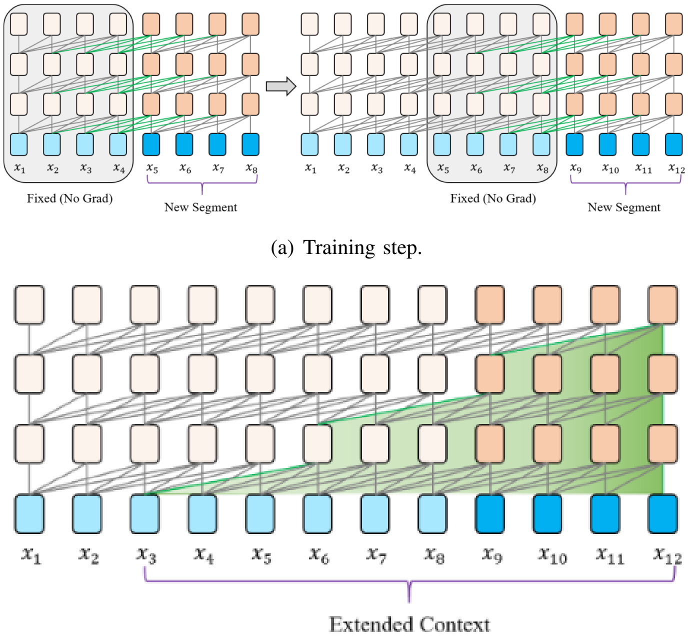
$$\begin{aligned} &\Rightarrow \underbrace{\mathbf{E}_{x_i}^\top \mathbf{W}_q^\top \mathbf{W}_{k,E} \mathbf{E}_{x_j}}_{\text{(a)}} + \underbrace{\mathbf{E}_{x_i}^\top \mathbf{W}_q^\top \mathbf{W}_{k,R} \mathbf{R}_{i-j}}_{\text{(b)}} \\ &+ \underbrace{u^\top \mathbf{W}_{k,E} \mathbf{E}_{x_j}}_{\text{(c)}} + \underbrace{v^\top \mathbf{W}_{k,R} \mathbf{R}_{i-j}}_{\text{(d)}} \end{aligned} \tag{2d}$$

c) Stochastic Temperature-Controlled Sampling:

In Transformer-XL's inference, stochastic temperature-controlled sampling blends Softmax With Temperature and Nucleus Sampling to enhance prediction diversity. Softmax With Temperature adjusts token probabilities via Equation 3, where τ modulates distribution spread. Nucleus Sampling targets the smallest token sets $V^{(p)}$ with cumulative probability exceeding threshold p for predicting subsequent tokens, expressed as $\sum_{z \in V^{(p)}} P(z|z_{1:i-1}) \geq p$, where $P(z|z_{1:i-1})$ denotes the probability of generating token z given tokens z from 1 to $i-1$.

$$p_i = \frac{e^{z_i/\tau}}{\sum_{j=1}^{K} e^{z_j/\tau}} \tag{3}$$

4) REMI: In this section, we begin by introducing the MIDI-like audio representation [32]. Subsequently, we delve into the REMI [21] audio representation and elucidate its distinctions from MIDI-like audio representation. Finally, we outline the processes involved in converting audio data into REMI audio representation.



(a) Training step.

(b) Inference step.

Fig. 5: The attention ideas of Transformer-XL with fixed length 4.

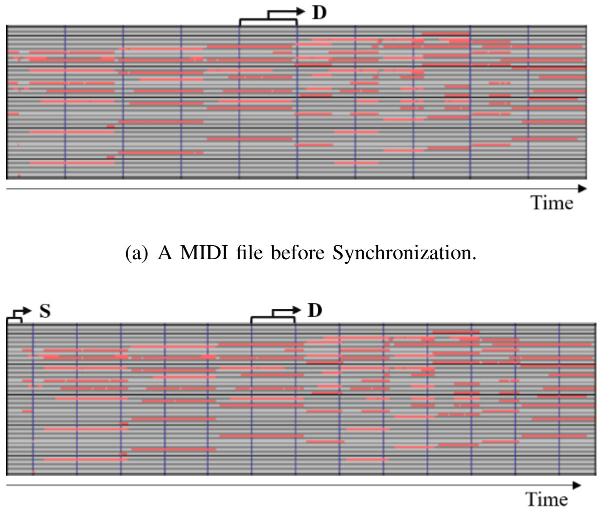
a) MIDI-like and REMI Audio Representation: The MIDI-like audio representation converts music data into a sequence of four distinct events: *NOTE-ON*, *NOTE-OFF*, *TIME-SHIFT*, and *VELOCITY*. A *NOTE-ON* event indicates the initiation of a note with a specified pitch, while a *NOTE-OFF* event signifies its termination. A *TIME-SHIFT* event advances the time step, and a *VELOCITY* event adjusts the volume of subsequent notes.

REMI [21] is an audio representation that allows for flexible local tempo changes while presenting challenges in controlling the rhythmic and harmonic structures of the music. It also takes advantage of MIDI-like audio representation by keeping the *NOTE-ON* and the *VELOCITY* events with the same concepts. REMI replaces the *NOTE-OFF* event with the *NOTE-DURATION* event, which gives how long a note should be played instead of recording when the note stops. In order to improve the structure of audio data, the *TIME-SHIFT* event is changed to the *POSITION* & *BAR* event that uses a bar as a unit to quantify the audio information. Moreover, the *TEMPO* and the *CHORD* events are added, where a *TEMPO* event gives the tempo information of the audio and a *CHORD* event represents a chord played in the audio by one of the 60 types of chords the authors set in advance.

b) Conversion Process: The processes involved in converting audio data into REMI audio representation can be divided into four parts as follows:

- Transcription: The Transcription process involves transforming the original audio file into the MIDI file format, which is a standard audio protocol, to capture the audio information. This process is automated using an API called Piano Transcription Inference [30].
- Synchronization: Following Transcription, the Synchronization process utilizes the MIDI file to compare it with the original audio file through beat tracking. This step allows for obtaining tempo and time-shifting information. The comparison between a MIDI file before and after Synchronization can be visualized in Figure 6. In the figure, differences in the distance between the blue lines, denoted as **D**, and the time-shifting at the beginning of the file, noted as **S**, can be observed.
- Analysis: The Analysis process uses 12 root notes to analyze the audio data after Synchronization as a way to manage the chord and melody information into events.
- Quantization: Subsequent to the aforementioned processes, the Quantization process ultimately quantizes all audio information into tokens based on events. The smallest unit that can be recorded is the 16th note in the audio. It is worth noting that the time signature of the audio must be a multiple of four.

Once the conversion processes are complete, a token dictionary will define the events represented by tokens in the REMI audio representation, as described in Section III-A4.



(a) A MIDI file before Synchronization.

(b) A MIDI file after Synchronization.

Fig. 6: The comparison of a MIDI file before and after Synchronization.

5) *YT-DLP:* YT-DLP [16] is a fork of YouTube-dl [15], a tool that interacts with the Youtube platform. YT-DLP offers a plethora of options for accessing information from YouTube, including Download options for downloading videos and Video Selection options for obtaining a list of videos based on specific criteria like file size and upload date.

IV. Design

The architecture of the proposed genre-conditioned piano music generation system (GENPIA) is illustrated in Figure 7. The system is divided into three phases, which include Data Collection, Data Pre-processing, and Model Training and Inference. In the Data Collection phase, audio data of different genres are gathered and labeled. Following this, the Data Pre-processing phase manages the audio data into the anticipated audio representation. Lastly, in the Model Training and Inference phase, the model acquires knowledge about the audio representation and produces the desired output.

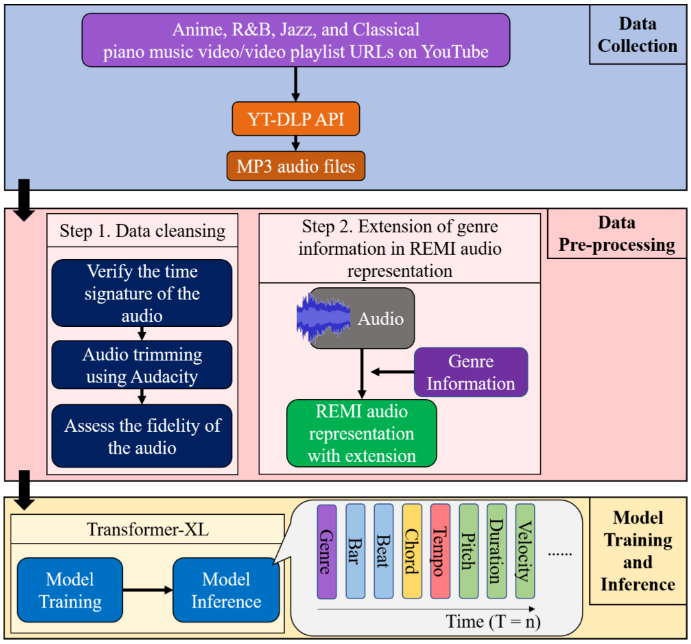

Anime, R&B, Jazz, and Classical piano music video/video playlist URLs on YouTube
Data Collection
YT-DLP API
MP3 audio files
Step 1. Data cleansing
Verify the time signature of the audio
Audio trimming using Audacity
Assess the fidelity of the audio
Step 2. Extension of genre information in REMI audio representation
Audio
Genre Information
REMI audio representation with extension
Data Pre-processing
Transformer-XL
Model Training
Model Inference
Genre
Bar
Beat
Chord
Tempo
Pitch
Duration
Velocity
......
Time (T = n)
Model Training and Inference

Fig. 7: The architecture of GENPIA.

1) *Data Collection:* On the YouTube platform, an increasing number of users are covering Anime and R & B music by piano. Classical and jazz piano music also enjoys wide recognition and popularity. Therefore, we select the *Anime*, *R & B*, *Jazz*, and *Classical* piano music from YouTube as our dataset.

In order to obtain the audio information from the YouTube platform, we employ the YT-DLP [16] API mentioned in Section III-A5. We utilize the Post-Processing options of YT-DLP to extract audio from video on YouTube. Before starting the training process, the audio data needs to be transcribed into a MIDI file, which is then used to create tokens. The model generates tokens as output, which are then converted back into MIDI files and further transformed into the MP3 format. Afterward, we provide either the URL of a YouTube page or a YouTube playlist URL from which we aim to extract the audio. As a labeling approach for the audio data, we organize the extracted audio into separate target folders based on music genre. Furthermore, we rename the filenames of the audio files to include their respective genres, following a specific format. This ensures that audio files for different genres are appropriately labeled and organized for further processing.

2) Data Pre-processing: Since we collect the dataset ourselves, the total number of audio files in our dataset may not be substantial. Hence, we opt to use the REMI [21] audio representation to pre-process our dataset. This allows us to avoid excessively short token lengths in each audio piece and provide an effective data structure for representing the audio data. Prior to converting the audio data into REMI audio representation, there are two preliminary steps: data cleansing and genre information addition. These steps are explained as follows.

a) Data Cleansing:

Three processes are involved in data cleansing, which is necessary due to the limitations of REMI audio representation.

- *Verify the time signature of the audio* : The time signature of the audio must be a multiple of four, as mentioned in Section III-A4b. Hence, it is necessary to implement a method to verify the compliance of the audio extracted by YT-DLP [16]. The Synchronization process of REMI [21] audio representation involves beat tracking, which is associated with the time signature of the audio. Following beat tracking, the Synchronization process utilizes time shifting to adjust the audio data. However, if the time signature is not a multiple of four, significant and abnormal time shifting occurs at the end of the audio data. Audio with abnormal time shifting noted as **ATS** is shown in Figure 8. Consequently, we remove all the audio with **ATS** after the Synchronization process of REMI audio representation.

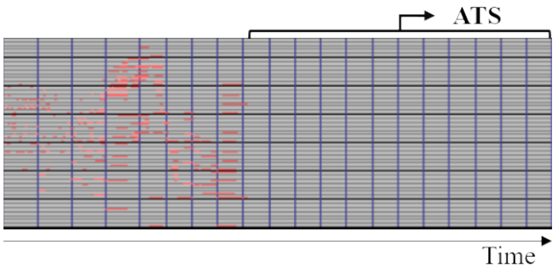


Fig. 8: A MIDI file with an abnormal time shifting.

- *Audio trimming using Audacity*: In this part, we segment each audio file into shorter clips, which helps reduce both training and inference time. This approach also directs the model's focus toward the most significant audio segments. Using Audacity [41], an audio editing software, we manually segment our dataset into shorter pieces that highlight the characteristic features of their respective music genres.
- *Assess the fidelity of the audio*: Several tokenization procedures involved in REMI [21] audio representation may lead to deviations from the original audio file. For instance, the Piano Transcription Inference [30] API, used in the Transcription process, may not always yield perfect accuracy and could introduce errors in pitch detection. Additionally, the smallest unit that can be recorded in the audio after pre-processing into REMI audio representation is the 16th note, potentially impacting the output. We have observed that frequent changes in tempo within the audio, particularly in Classical and Jazz genres, can lead to inaccuracies during processing by REMI audio representation. To address these issues, each pre-processed audio undergoes manual fidelity assessment through careful listening. Any audio deemed unqualified is removed from the dataset.

The number of audio files in each genre after each process in data cleansing is shown in Table I. The result of data cleansing contains 422 Anime audio clips, 462 R & B audio clips, 244 Jazz audio clips, and 189 Classical audio clips in our dataset. Additionally, the audio clips in our dataset have a duration that ranges from 11 to 37 seconds.

TABLE I: The number of audio in each genre after each process in data cleansing

Process	**Number of Audio (after the specified process)**			
	Anime	**R&B**	**Jazz**	**Classical**
Extracted by YT-DLP [16]	710	405	243	395
Verify the time signature of the audio	477	342	193	192
Audio trimming using Audacity [41]	482	480	341	300
Assess the fidelity of the audio	422	462	244	189

b) Extension of Genre Information in REMI Audio Representation:

To include the genre label information into the REMI [21] audio representation, we adopt a similar approach to that of EMOPIA [22], which adds emotion label information into the compound word representation(CP) [19]. By utilizing the genre label information from the filenames during the Quantization process of REMI audio representation, we provide a novel type of token that is specifically associated with the genre of an audio file. Since the pre-processed audio tokens are sequentially dependent due to their temporal relationship, the genre-related token is added at the beginning of a sequence of audio tokens as an extension of REMI audio representation. Figure 9 illustrates the process of adding genre label information to the REMI audio representation.

3) Model Training and Inference: The Transformer-XL [9] model, as explained in Section III-A3a, employs the segment-level recurrence technique to address the context fragmentation problem and extend the dependency length of input sequences. When shifting the condition of music generation from emotion to genre, a model with a higher input dependency length becomes more suitable for learning audio patterns across various music genres. Hence, we opt for the Transformer-XL model in our approach.

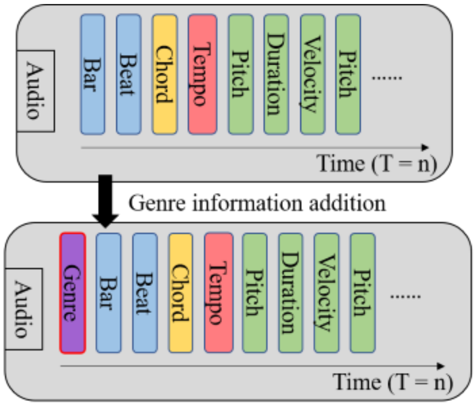


Fig. 9: Adding genre label information into REMI audio representation.

a) Model Training:

Our model configuration is based on the Transformer-XL [9] applied in CP [19] with a fixed window size of 512. However, after pre-processing, the average token length of our dataset, converted into REMI [21] audio representation with extension, is 776. To ensure that all tokens are considered within the fixed window size of 512 in the Transformer-XL model, we set the group size to 2, allowing a maximum input length of 1024. Each input token's ground truth is defined as the token that follows it in the sequence. Additionally, besides the training data, the corresponding token dictionary is required as input to the model to determine the number of token types in the training data, aiding the token embedding process.

The loss function used in the Transformer-XL model is the same as in the Transformer [43] model, namely, the cross-entropy loss [40]. In the task of music generation, the model's objective is to create novel audio output based on the knowledge acquired during training. Therefore, there is no strict requirement for employing a test dataset. Including a test dataset in model evaluation may sometimes result in a high calculated loss. Additionally, training the model to an excessively low loss might lead it to simply reproduce the training data. Therefore, we set our loss target value similar to prior works in music generation [19, 22], which share a similar research objective to ours.

b) Model Inference:

To generate music of a specific genre, we leverage the way the genre information is extended in REMI [21] audio representation, which inserts the genre label information at the beginning of a sequence of audio tokens. When the model is provided with the desired music genre, this genre information is converted into a genre token by utilizing the corresponding token dictionary. By incorporating both genre tokens and initial tokens (i.e., bar and beat tokens) into the model, it acquires prior knowledge about the genre. Subsequently, the model generates the following content based on the target genre information using Stochastic Temperature-Controlled Sampling until it encounters an end-of-file (EOF) token. Figure 10 illustrates this generating process. After the sampling process, we utilize the corresponding token dictionary to convert audio tokens into events. These events are then used to generate MIDI files. Finally, we apply the Salamander Grand Piano [1] sound font to achieve the desired piano timbre

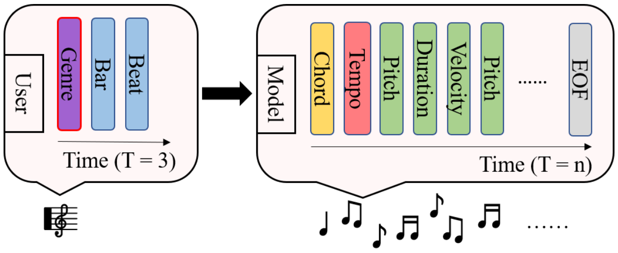


Fig. 10: Generation process illustration.

V. Experiments and Results

In this section, we employ the method of EMOPIA [22], which represents the state-of-the-art approach for Russell's 4Q [38] emotion-conditioned piano music generation, to accomplish genre-conditioned piano music generation. This serves as a benchmark to evaluate and compare with our work. Subsequent sections offer detailed descriptions of the external dataset utilized in the experiment, the configuration of the experimental environment, the evaluation metrics employed, and a comprehensive analysis of the experimental results.

A. External Dataset

It's widely recognized that models tend to achieve better results when trained on larger datasets. The approach outlined in EMOPIA [22] attains its highest performance by utilizing the Ailabs.tw 1K7 [19] dataset for pre-training. Therefore, we also leverage the Ailabs.tw 1K7 dataset for pre-training purposes. The Ailabs.tw 1K7 dataset comprises 1748 Pop piano performances sourced from the Internet, with an average audio duration of around 4 minutes per performance. Importantly, the time signature of all audio files in the dataset adheres to a multiple of four. This dataset provides a rich source of diverse piano performances to enhance our model's training process.

B. Experimental Environment Configuration

The experimental setup includes both hardware and software components. We employ two sets: Set 1 for data collection and Set 2 for data pre-processing, model training, and inference. Set 1 features an Intel(R) Core(TM) i7-10700 CPU with 8 cores, 32 GB of RAM, an NVIDIA GeForce RTX 2060 with 6GB VRAM, and operates on Windows 11x64. Set 2 is equipped with an Intel(R) Xeon(R) Gold 5218 CPU with 12 cores, 64 GB of RAM, an NVIDIA Quadro RTX 6000 with 24GB VRAM, and runs on Ubuntu 22.04.1 LTS. During model training, Adam optimizer [27] is utilized, the batch size is set to 4, the learning rate is set to 0.0001, and the random seed is fixed to 2222. The training process aims to attain a target loss of 0.2, consistent with the loss value used in CP [19]. For the pre-training approach, the model initially undergoes training with the Ailabs.tw 1K7 [19] dataset until a loss of 0.3 is achieved, consistent with the loss value used in EMOPIA [22]. Subsequently, the pre-trained model is further trained with our dataset until a loss of 0.2 is attained. Notably, during pre-training, the genre token in the dataset is treated as *ignore*. To ensure an output with a desired level of diversity, we set the parameter values of τ to 1.2 and p to 0.9 in Stochastic Temperature-Controlled Sampling, which are the same as in CP [19].

C. Evaluation Metrics

The most effective method for evaluating a music generation model currently remains listening tests. Therefore, we employ subjective evaluation metrics in our research, which include *Richness*, *Humanness*, *Correctness*, *Structureness*, and *Genre Similarity*. While *Richness*, *Humanness*, *Correctness*, and *Structureness* are utilized in prior works [19, 22], *Genre Similarity* is a new addition to support our research. These subjective evaluation metrics are described as follows.

1) *Richness*: Evaluates the diversity and attraction level of the output audio.
2) *Humanness*: Assesses the level of similarity between the output audio and the audio performed by a human.
3) *Correctness*: Measures the absence of perceived incorrect notes in the output audio based on music theory.
4) *Structureness*: Determines the level of presence of structural patterns such as recurring themes or melodic motifs in the output audio.
5) *Genre Similarity*: Measures the match level of the output audio with the target genre.

D. Experimental Results

This section presents the experimental results of four different types of models: EMOPIA [22] and GENPIA with and GENPIA without pre-training. We obtained the results utilizing a listening questionnaire designed by ourselves and divided into four parts, each corresponding to one of the four music genres. Each part contains nine piano music clips, with a duration less than 30 seconds, comprising one demo clip and eight evaluation clips. Despite the clips being slightly shorter than 30 seconds, it is sufficient to evaluate their structural coherence. The demo clip is a randomly selected audio file from our dataset that corresponds to the specific music genre. The purpose of selecting the demo clip is to ensure that listeners are more familiar with a particular music genre. The eight clips to be evaluated in each part are composed of two audio outputs of four different types of models. In summary, there are two audio outputs for each type of model in each part, resulting in a total of 32 music clips for evaluation in the listening questionnaire. Listeners evaluate the music clip based on the metrics mentioned in Section V-C rating them on a five-point Likert scale [47]. Typically, it takes about 30 minutes to finish the listening questionnaire.

The survey included 55 participants, mainly from the school's wind band, choir, and guitar clubs. Among them, 4 had no prior music instrument learning experience, 4 had less than one year of experience, 15 had 1 to 5 years, 26 had 6 to 10 years, and 6 had over 11 years of experience. Finally, we demonstrate the mean outcomes of four types of music genres, which is the average results of 55 subjects rounded to the third decimal place. In our work (**GENPIA**), the metrics are represented as follows: *Richness* (**R**), *Humanness* (**H**), *Correctness* (**C**), *Structureness* (**S**), and *Genre Similarity* (**G**).

TABLE II: Survey results of the Anime music genre. **Bold with Underline** indicates best, and Underline indicates second.

Anime					
Method	**Subjective Metrics**				
	R	**H**	**C**	**S**	**G**
EMOPIA	3.173	2.964	2.827	2.873	2.482
GENPIA	3.927	3.655	3.682	3.6	3.245
EMOPIA w/ pre-training	3.264	3.009	2.864	2.945	2.873
GENPIA w/ pre-training	**4.018**	**3.827**	**3.718**	**3.836**	**3.782**

According to Table II GENPIA with pre-training exhibits superior performance in the Anime music genre compared to other methods. It outperforms other approaches across all subjective metrics. Moreover, even without pre-training, GENPIA still achieves the second-highest rankings in all subjective metrics.

TABLE III: Survey results of the R & B music genre. **Bold with Underline** indicates best, and Underline indicates second.

R & B					
Method	**Subjective Metrics**				
	R	**H**	**C**	**S**	**G**
EMOPIA	2.882	2.945	2.9	2.655	2.582
GENPIA	**3.445**	**3.636**	**3.718**	**3.464**	**3.4**
EMOPIA w/ pre-training	3.009	3.263	3.136	3.09	3.027
GENPIA w/ pre-training	3.255	3.6	3.227	3.427	3.3

According to Table III, it can be seen that GENPIA without pre-training surpasses other methods in terms of all the subjective metrics within the R & B music genre. With pre-training applied, GENPIA consistently achieves the second-highest ranking across all subjective metrics.

TABLE IV: Survey results of the Jazz music genre. **Bold with Underline** indicates best, and Underline indicates second.

Jazz					
Method	**Subjective Metrics**				
	R	**H**	**C**	**S**	**G**
EMOPIA	2.6	2.536	2.564	2.564	2.509
GENPIA	3.409	**3.627**	**3.618**	**3.745**	**3.573**
EMOPIA w/ pre-training	**3.418**	3.127	2.791	3.118	3.055
GENPIA w/ pre-training	3.364	3.1	3.055	3.155	3.1

As shown in Table IV, GENPIA without pre-training outperforms other methods in terms of subjective metrics *Humanness*, *Correctness*, *Structureness*, and *Genre Similarity*. Additionally, it obtains the second place in the subjective metric *Richness*. EMOPIA [22] with pre-training obtains first place and second place in subjective metrics *Richness* and

Humanness, respectively. In the context of pre-training utilization, GENPIA achieves the second position across subjective metrics *Correctness*, *Structureness*, and *Genre Similarity*. Furthermore, GENPIA with and without pre-training demonstrates only a minor disparity compared to EMOPIA with pre-training in subjective metric *Humanness* and *Richness*, respectively.

TABLE V: Survey results of the Classical music genre. **Bold with Underline** indicates best, and Underline indicates second.

Classical					
Method	**Subjective Metrics**				
	R	**H**	**C**	**S**	**G**
EMOPIA	2.8	2.845	2.591	2.682	2.7
GENPIA	2.8	3.555	3.582	3.391	**3.664**
EMOPIA w/ pre-training	**3.191**	2.918	2.909	2.927	2.845
GENPIA w/ pre-training	3.164	**3.664**	**3.836**	**3.436**	3.391

Table V highlights that GENPIA with pre-training outperforms other methods in subjective metrics *Humanness*, *Correctness*, and *Structureness*, and ranks second in both subjective metrics *Richness* and *Genre Similarity*. Even without pre-training, GENPIA outperforms other methods in subjective metric *Genre Similarity* and achieves second place in terms of subjective metrics *Humanness*, *Correctness*, and *Structureness*. Additionally, EMOPIA [22] acquires the first place in terms of subjective metric *Richness*. However, when pre-training is applied, GENPIA has only a minor disparity compared to EMOPIA with pre-training in terms of subjective metric *Richness*.

Based on the aforementioned findings, it can be concluded that GENPIA with pre-training demonstrates the overall best performance within the Anime and Classical music genres. Conversely, GENPIA without pre-training has the best overall performance within the R & B and Jazz music genres. These outcomes indicate that GENPIA is well-suited for our research objective. Moreover, when pre-training is utilized, GENPIA has the potential to enhance performance within certain music genres. We believe the reason behind this observation is that the music genres included in the pre-training data closely align with the specific music genres we have chosen.

VI. Conclusion

This study proposes GENPIA, a system for genre conditioned piano music generation. The system involves the collection and labeling of a custom music dataset. To address the limitations of REMI [21] audio representation, we perform data cleansing techniques before converting the audio files into REMI audio representation with extension. To better construct the audio data with its music genre, we extend the REMI audio representation to incorporate genre information in data preprocessing. Our model utilizes Transformer-XL [9], which can better capture long-range dependencies, a crucial capability for learning audio patterns across diverse music genres. Additionally, the implemented genre information addition supports model inference. According to the survey results, our approach is better than the EMOPIA method [22] in the task of genre-conditioned piano music generation using our custom music dataset. Furthermore, by incorporating pre-training techniques, our approach demonstrates improved performance within the Anime music genre in all the subjective metrics, and within the Classical music genre in the subjective metrics of *Richness*, *Humanness*, *Correctness*, and *Structureness*.

References

[1] Salamander grand piano. https://freepats.zenvoid.org/Piano/acoustic-grand-piano.html.

[2] Denk T. I. Borsos Z. Engel J. Verzetti M. Caillon A. ... Frank C Agostinelli, A. Musiclm: Generating music from text. *arXiv preprint arXiv:2301.11325*, 2023.

[3] Rami Al-Rfou, Dokook Choe, Noah Constant, Mandy Guo, and Llion Jones. Character-level language modeling with deeper self-attention. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 3159–3166, 2019.

[4] Kiros J. R. Hinton G. E. Ba, J. L. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[5] Salamon J. Tierney M. Mauch M. Cannam C. Bello J. P. Bittner, R. M. Medleydb: A multitrack dataset for annotation-intensive mir research. In *ISMIR*, volume 14, pages 155–160, 2014.

[6] Huang Y. H. Hsiao W. Y. Yang Y. H. Chen, Y. H. Automatic composition of guitar tabs by transformers and groove modeling. *arXiv preprint arXiv:2008.01431*, 2020.

[7] Chung Y. Lee S. Jeon J. Kwon T. Nam J. Choi, E. Ym2413-mdb: A multi-instrumental fm video game music dataset with emotion annotations. *ArXiv*, abs/2211.07131, 2022.

[8] Hawthorne C. Simon I. Dinculescu M. Engel J. Choi, K. Encoding musical style with transformer autoencoders. In *ICML*, pages 1899–1908. PMLR, 2020.

[9] Yang Z. Yang Y. Carbonell J. Le Q. V. Salakhutdinov R. Dai, Z. Transformer-xl: Attentive language models beyond a fixed-length context. *arXiv preprint arXiv:1901.02860*, 2019.

[10] Benzi K. Vandergheynst P. Bresson X. Defferrard, M. Fma: A dataset for music analysis. *arXiv preprint arXiv:1612.01840*, 2016.

[11] Mao H. H. Li Y. E. Cottrell G. W. McAuley J. Donahue, C. Lakhnes: Improving multi-instrumental music generation with cross-domain pre-training. *arXiv preprint arXiv:1907.04868*, 2019.

[12] Jeff Ens and Philippe Pasquier. Mmm: Exploring conditional multi-track music generation with the transformer. *arXiv preprint arXiv:2008.06048*, 2020.

[13] Tatar K. Thorogood M. Pasquier P. Fan, J. Ranking-based emotion recognition for experimental music. In *ISMIR*, volume 2017, pages 368–375, 2017.

[14] Thorogood M. Pasquier P. Fan, J. Emo-soundscapes: A dataset for soundscape emotion recognition. In *2017 17th ACII*, pages 196–201. IEEE, 2017.

[15] github. Youtube-dl. https://github.com/ytdl-org/youtube-dl/, 2021.

[16] github. Yt-dlp. https://github.com/yt-dlp/yt-dlp/, 2023.

[17] Pouget-Abadie J. Mirza M. Xu B. Warde-Farley D. Ozair S. ... Bengio Y. Goodfellow, I. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

[18] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 1997.

[19] Liu J. Y. Yeh Y. C. Yang Y. H. Hsiao, W. Y. Compound word transformer: Learning to compose full-song music over dynamic directed hypergraphs. In *AAAI Conference on AI*, volume 35, pages 178–186, 2021.

[20] Vaswani A. Uszkoreit J. Shazeer N. Hawthorne C. Dai A. M. ... Eck D. Huang, C. Z. A. Music transformer: Generating music with long-term structure (2018). *arXiv preprint arXiv:1809.04281*, 2018.

[21] Yu-Siang Huang and Yi-Hsuan Yang. Pop music transformer: Beat-based modeling and generation of expressive pop piano compositions. In *Proceedings of the 28th ACM Multimedia*, pages 1180–1188, 2020.

[22] Ching J. Doh S. Kim N. Nam J. Yang Y. H. Hung, H. T. Emopia: A multi-modal pop piano dataset for emotion recognition and emotion-based music generation. *arXiv preprint arXiv:2108.01374*, 2021.

[23] Gu S. Turner R. E. Eck D. Jaques, N. Generating music by fine-tuning recurrent neural networks with reinforcement learning. 2016.

[24] Xia G. G. Carlton D. B. Anderson C. N. Miyakawa R. H. Jiang, J. Transformer vae: A hierarchical model for structure-aware and interpretable music representation learning. In *ICASSP*, pages 516–520. IEEE, 2020.

[25] Leslie Pack Kaelbling, Michael L Littman, and Andrew W Moore. Reinforcement learning: A survey. *Journal of AI research*, 4:237–285, 1996.

[26] Vyas A. Pappas N. Fleuret F. Katharopoulos, A. Transformers are rnns: Fast autoregressive transformers with linear attention. In *ICML*, pages 5156–5165. PMLR, 2020.

[27] Ba J. Kingma, D. P. Adam: A method for stochastic optimization. *CoRR*, 2015.

[28] Welling M. Kingma, D. P. An introduction to variational autoencoders. *Foundations and Trends® in Machine Learning*, 12(4):307–392, 2019.

[29] Li B. Chen J. Wang Y. Kong, Q. Giantmidi-piano: A large-scale midi dataset for classical piano music. *Trans. Int. Soc. Music. Inf. Retr.*, 5:87–98, 2020.

[30] Li B. Song X. Wan Y. Wang Y. Kong, Q. High-resolution piano transcription with pedals by regressing onset and offset times. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:3707–3717, 2021.

[31] Midjourney.com. Midjourney. https://www.midjourney.com.

[32] Simon I. Dieleman S. Eck D. Simonyan K. Oore, S. This time with feeling: Learning expressive musical performance. *Neural Computing and Applications*, 32:955–967, 2020.

[33] OpenAI. Chatgpt. https://openai.com/research/chatgpt, 2021. Accessed: June 1, 2023.

[34] Malheiro R. Rocha B. Oliveira A. P. Paiva R. P. Panda, R. E. S. Multi-modal music emotion recognition: A new dataset, methodology and comparative analysis. In *10th CMMR 2013*, pages 570–582, 2013.

[35] C Payne. Musenet. openai blog, 2019.

[36] Engel J. Raffel C. Hawthorne C. Eck D. Roberts, A. A hierarchical latent vector model for learning long-term structure in music. In *ICML*, pages 4364–4373. PMLR, 2018.

[37] Hinton G. E. Williams R. J. Rumelhart, D. E. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.

[38] James A Russell. A circumplex model of affect. *Journal of personality and social psychology*, 39(6):1161, 1980.

[39] Kamal O. Jin Z. Schölkopf B. Schneider, F. Moûsai: Text-to-music generation with long-context latent diffusion. *arXiv preprint arXiv:2301.11757*, 2023.

[40] C. E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27(3):379–423, 1948.

[41] Team Audacity. Audacity. https://www.audacityteam.org/, 2000.

[42] Luca Turchet, Mathieu Lagrange, Cristina Rottondi, György Fazekas, Nils Peters, Jan Østergaard, Frederic Font, Tom Bäckström, and Carlo Fischione. The internet of sounds: Convergent trends, insights, and future directions. *IEEE Internet of Things Journal*, 10(13):11264–11292, 2023.

[43] Shazeer N. Parmar N. Uszkoreit J. Jones L. Gomez A. N. Polosukhin I. Vaswani, A. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[44] Chen K. Jiang J. Zhang Y. Xu M. Dai S. ... Xia G. Wang, Z. Pop909: A pop-song dataset for music arrangement generation. In *ISMIR*, 2020.

[45] Yang Y. H. Wu, S. L. The jazz transformer on the front line: Exploring the shortcomings of ai-composed music through quantitative measures. *arXiv preprint arXiv:2008.01307*, 2020.

[46] Hu C. Wang Y. Hu X. Zhu J. Wu, J. A hierarchical recurrent neural network for symbolic melody generation. *IEEE transactions on cybernetics*, 50(6):2749–2757, 2019.

[47] K. L. Wuensch. What is a likert scale? and how do you pronounce'likert?'. *East Carolina University*, 4, 2005.

[48] Le Cun Yan, B Yoshua, and H Geoffrey. Deep learning. *nature*, 521(7553):436–444, 2015.

[49] Chou S. Y. Yang Y. H. Yang, L. C. Midinet: A convolutional generative adversarial network for symbolic-domain music generation. *arXiv preprint arXiv:1703.10847*, 2017.